

TUGAS AKHIR - TJ141502

**PEMANTAUAN PELANGGARAN KECEPATAN
KENDARAAN MENGGUNAKAN DUA SENSOR KAMERA**

Aldiansyah Ramadhan
NRP 07211440000040

Dosen Pembimbing
Dr. Eko Mulyanto Yuniarno, ST., MT.
Arief Kurniawan, ST., MT.

DEPATERMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TJ141502

**PEMANTAUAN PELANGGARAN KECEPATAN
KENDARAAN MENGGUNAKAN DUA SENSOR KAMERA**

Aldiansyah Ramadhan
NRP 07211440000040

Dosen Pembimbing
Dr. Eko Mulyanto Yuniarno, ST., MT.
Arief Kurniawan, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TJ141502

***VEHICLE SPEED VIOLATION MONITORING USING
TWO CAMERA SENSORS***

Aldiansyah Ramadhan
NRP 0721144000040

Advisor
Dr. Eko Mulyanto Yuniarno, ST., MT.
Arief Kurniawan, ST., MT.

Departement of Computer Engineering
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Pemantauan Pelanggaran Kecepatan Kendaraan Menggunakan Dua Sensor Kamera”** adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2018



Aldiansyah Ramadhan
NRP. 0721144000040

LEMBAR
PENGESAHAN

LEMBAR PENGESAHAN

PEMANTAUAN PELANGGARAN KECEPATAN KENDARAAN MENGUNAKAN DUA SENSOR KAMERA

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Aldiansyah Ramadhan (NRP: 07211440000040)

Tanggal Ujian : 29 Juni 2018

Periode Wisuda : September 2018

Disetujui oleh:

Dr. Eko Mulyanto Yuniarno, ST., MT.
NIP: 196806011995121009

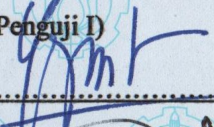
(Pembimbing I)



(Pembimbing II)

Arief Kurniawan, ST., MT.
NIP: 197409072002121001

(Penguji I)



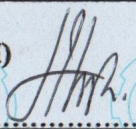
Prof. Dr. Ir. Yoyon K. Suprpto, M.Sc.
NIP: 195409251978031001

(Penguji II)



Ahmad Zaini, ST., M.Sc.
NIP: 197504192002121003

(Penguji III)



Muhtadin, ST., MT.
NIP: 198106092009121003

Mengetahui
Kepala Departemen Teknik Komputer

Dr. I Ketut Eddy Purnama, ST., MT.
NIP. 196907301995121001

DEPARTEMEN
TEKNIK KOMPUTER

ABSTRAK

Nama Mahasiswa : Aldiansyah Ramadlan
Judul Tugas Akhir : Pemantauan Pelanggaran Kecepatan Kendaraan Menggunakan Dua Sensor Kamera
Pembimbing : 1. Dr. Eko Mulyanto Yuniarno, ST., MT.
2. Arief Kurniawan, ST., MT.

Berdasarkan data kepolisian, tingkat jumlah kecelakaan pada tahun 2016 mencapai 105 ribu jiwa yang dikarenakan banyaknya jumlah pelanggaran kecepatan [1]. Sistem pemantauan kecepatan saat ini yang memanfaatkan *IP-Camera* belum bisa menangkap citra pengemudi secara otomatis jika terjadi pelanggaran kecepatan sehingga masih dilakukan secara manual. Pemasangan *IP-Camera* yang membutuhkan infrastruktur tetap dapat membatasi petugas dalam memantau pelanggaran kecepatan. Sehingga pada penelitian ini dikerjakan suatu sistem yang dapat memantau pelanggaran kecepatan dengan menggunakan dua *IP-Camera*. Sistem ini dapat menentukan kendaraan yang telah terdeteksi melanggar kecepatan, metode yang digunakan yaitu menggunakan *matching-based* untuk penjejakan. Penggunaan dua kamera bertujuan untuk mendapatkan citra melanggar kecepatan dari dua posisi yang berbeda, metode untuk kalibrasi kedua kamera menggunakan metode *projective transformation*. Berdasarkan hasil pengujian pada sistem, menunjukkan bahwa sistem ini dapat menghitung kecepatan kendaraan yang terdeteksi dengan akurasi terbaik yaitu sebesar 97.01% pada kondisi tidak ada bayangan dengan jumlah *frame rate* 30fps dan dengan akurasi 83.71% pada kondisi ada bayangan. Sistem ini juga dapat menentukan jenis kendaraan mobil atau motor yang terdeteksi melanggar dengan akurasi sebesar 89.62%. Tingkat akurasi yang dicapai dalam mengambil citra pengemudi yang melanggar batas kecepatan yaitu sebesar 97.71%.

Kata Kunci : Dua *IP-Camera*, *Background Reconstruction*, *Background Subtraction*, Penghitungan kecepatan Kendaraan, *Matching-based*, *Projective Transformation*

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Aldiansyah Ramadhan
Title : *Vehicle Speed Violation Monitoring Using
Two Camera Sensors*
Advisors : 1. Dr. Eko Mulyanto Yuniarno, ST., MT.
2. Arief Kurniawan, ST., MT.

Based on police data, the number of accidents in 2016 reached 105 thousand inhabitants due to the number of violations of the speed. The current speed monitoring system that utilizes IP-Camera has not been able to capture the image of the rider automatically in case of speed violation so it is still done manually. Installation of IP-Camera that requires fixed infrastructure may limit officers to monitor speed violations. So in this research is done a system that can monitor speed violation by using two IP-Camera. This system can determine the vehicle that has been detected breaking speed, the method used is using matching-based for tracking. The use of two cameras aims to obtain an image of speed breakers from two different positions, a method for calibrating both cameras using the projective transformation method. Based on the results of testing on the system, shows that this system can calculate the speed of the vehicle is detected with an accuracy of 97.01% on the condition there is no shadow with the number of frame rate 30fps and with the accuracy 83.86% on the condition there is a shadow. This system can also determine the type of car or motor vehicle detected breaking with an accuracy of 89.62%. Accuracy rate achieved in taking the image of motorists who violated the speed limit that is equal to 97.71 %.

Keywords : Two IP-Camera, Background Reconstruction, Background Subtraction, Vehicle speed calculation, Matching-based, Projective Transformation

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas segala karunia-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Pemantauan Pelanggaran Kecepatan Kendaraan Menggunakan Dua Sensor Kamera**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. I Ketut Eddy Purnama, ST., MT. selaku Ketua Departemen Teknik Komputer, Fakultas Teknik Elektro, Institut Teknologi Sepuluh Nopember.
3. Secara khusus penulis mengucapkan terima kasih kepada Bapak Dr. Eko Mulyanto Yuniarno, ST. dan Bapak Arief Kurniawan, ST., MT. atas bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman dari Teknik Komputer, Laboratorium B201, B401 dan Computer Vision.
6. Zulfa Anis Labiba selaku seseorang yang selalu menemani saya dalam pengambilan data dan pengerjaan penelitian ini.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2018

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xv
NUMENKLATUR	xvii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	2
1.5 Sistematika Penulisan	3
1.6 Relevansi	4
2 TINJAUAN PUSTAKA	5
2.1 Pengukuran Kecepatan	5
2.2 <i>IP Camera</i>	6
2.3 <i>Euclidean Distance</i>	6
2.4 <i>Projective Transformation</i>	6
2.5 <i>Background Reconstruction</i>	9
2.6 <i>Background Subtraction</i>	11
2.7 <i>Thresholding</i>	12
2.8 <i>Image Cropping</i>	13
2.9 <i>Algoritma Matching Based</i>	13
2.10 Penelitian Terkait Perhitungan Jumlah Kendaraan dan Deteksi Kecepatan Kendaraan	14

3	DESAIN DAN IMPLEMENTASI SISTEM	17
3.1	Desain Sistem	17
3.2	Work Flow	21
3.3	Akuisisi Data	21
3.3.1	Persiapan dan Peletakan Kamera	21
3.3.2	Pemasangan Penanda	22
3.3.3	Lokasi Akuisisi Data	27
3.3.4	IP Camera yang Digunakan	28
3.3.5	Pengambilan Data Video	28
3.4	Kalibrasi Kedua Kamera	31
3.4.1	Ekstraksi Citra	31
3.4.2	Projective Transformation	32
3.5	Penjejakan Kendaraan	35
3.5.1	Seleksi Area	36
3.5.2	Pembuatan citra Latar	37
3.5.3	Substraksi Citra Latar	39
3.5.4	Substraksi Citra Biner	42
3.5.5	Operasi Morfologi	44
3.5.6	Deteksi Kendaraan	46
3.5.7	Deteksi Jenis Kendaraan	50
3.5.8	Proses Penjejakan Kendaraan	51
3.6	Perhitungan Kecepatan Kendaraan	54
3.7	Pelanggaran Kecepatan	56
3.7.1	Tangkap Citra Pertama Pelanggar	56
3.7.2	Tangkap Citra Kedua Pelanggar	58
3.8	Antarmuka Pengguna Berbasis Desktop	62
4	PENGUJIAN DAN ANALISA	65
4.1	Pengujian dengan Perbedaan Jumlah <i>Frame Rate</i> . .	65
4.2	Pengujian dengan Perbedaan Kategori Bayangan . .	70
4.3	Pengujian Jenis Kendaraan	71
4.4	Pengujian dengan Perbedaan Pemilihan ROI	72
4.5	Pengujian Pelanggaran Kecepatan	74
4.6	Durasi Pengiriman Data Pelanggaran Kecepatan . .	79
4.7	Pengujian Kecepatan dengan Menggunakan Persa- maan Regresi	81

5	PENUTUP	83
5.1	Kesimpulan	83
5.2	Saran	83
	DAFTAR PUSTAKA	85
	Biografi Penulis	89

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

2.1	Citra masukan dari <i>background reconstruction</i>	10
2.2	Hasil proses <i>background reconstruction</i>	11
2.3	Citra sebelum dan sesudah proses <i>Background subtraction</i>	11
2.4	Citra sebelum dan sesudah proses <i>binary thresholding</i>	12
2.5	Citra masukan dan citra keluaran proses <i>image cropping</i>	13
2.6	<i>complete bi-partite graph</i>	14
3.1	Diagram alir sistem pemantauan pelanggaran kecepatan.	17
3.2	Diagram alir metodologi penelitian untuk kalibrasi kamera.	18
3.3	Diagram alir metodologi penelitian untuk pemantauan pelanggaran kecepatan.	19
3.4	Skema kamera pertama.	21
3.5	Skema kedua kamera.	22
3.6	Skema kamera 1.	23
3.7	Pemasangan penanda pada pinggir jalan.	23
3.8	Pemasangan penanda pada pinggir jalan.	24
3.9	Pemasangan penanda pada pinggir jalan.	25
3.10	Pemasangan penanda pada pinggir jalan.	26
3.11	Pemasangan penanda pada pinggir jalan.	26
3.12	Lokasi akusisi data.	27
3.13	Jenis kamera yang digunakan.	28
3.14	Perekaman video pada pagi hari.	30
3.15	Perekaman video pada siang hari	31
3.16	Perekaman video pada sore hari.	31
3.17	Proses ekstraksi citra 1.	32
3.18	Proses ekstraksi citra 2.	32
3.19	Diagram alir proses <i>perspective transform</i>	33
3.20	Proses <i>perspective transform</i> pada citra 1	34
3.21	Proses <i>perspective transform</i> pada citra 2	34
3.22	Diagram alir proses <i>perspective transform</i>	35
3.23	Proses Seleksi Area Menggunakan <i>Bitwise And</i>	36

3.24	Diagram alir proses pembuatan citra latar.	37
3.25	Proses konversi ruang warna <i>bgr</i> ke <i>grayscale</i>	38
3.26	Citra hasil proses <i>background reconstruction</i> dengan jumlah citra masukan sebanyak 75 citra.	38
3.27	Citra hasil proses <i>background reconstruction</i> dengan jumlah citra masukan sebanyak 150 citra.	39
3.28	Citra hasil proses <i>background reconstruction</i> dengan jumlah citra masukan sebanyak 300 citra.	39
3.29	Diagram alir proses pembuatan citra latar depan (<i>Background subtraction</i>).	40
3.30	Citra hasil <i>Background subtraction</i>	41
3.31	Citra hasil proses <i>background subtraction</i> menggunakan citra latar dengan jumlah citra masukan sebanyak 75 citra.	41
3.32	Citra hasil proses <i>background subtraction</i> menggunakan citra latar dengan jumlah citra masukan sebanyak 150 citra.	42
3.33	Citra hasil proses <i>background subtraction</i> menggunakan citra latar dengan jumlah citra masukan sebanyak 300 citra.	42
3.34	Citra hasil proses <i>thresholding</i> dengan nilai ambang sepuluh.	43
3.35	Citra hasil proses <i>thresholding</i> dengan nilai ambang dua puluh.	43
3.36	Citra hasil proses <i>thresholding</i> dengan nilai ambang tiga puluh.	44
3.37	Citra hasil proses <i>thresholding</i>	45
3.38	Citra hasil proses <i>erosion</i> dengan nilai iterasi sejumlah tiga.	45
3.39	Citra hasil proses <i>dilation</i> dengan nilai iterasi sejumlah lima.	45
3.40	Citra hasil proses <i>erosion</i> dan <i>dilation</i> sebanyak lima kali.	46
3.41	Citra hasil proses operasi morfologi dan Deteksi kontur.	47
3.42	Menggambar <i>circle</i> pada objek yang terdeteksi . . .	49
3.43	Diagram alir menggambar <i>circle</i> dan <i>rectangle</i> pada objek yang terdeteksi.	49

3.44	Diagram alir deteksi jenis kendaraan pada objek yang terdeteksi.	50
3.45	Pencarian hubungan antar <i>frame</i>	51
3.46	Graf hubungan antar frame.	52
3.47	Hasil penjejakkan kendaraan bermotor.	53
3.48	Batas awal dan batas akhir dalam penentuan jarak tempuh	54
3.49	Diagram alir proses penjejakkan.	55
3.50	Percobaan Pelanggaran Kecepatan.	57
3.51	Tangkap citra pelanggar kecepatan	57
3.52	Ilustrasi pemasangan penanda pipa	58
3.53	Diagram alir proses penjejakkan.	59
3.54	Citra pelanggar hasil tangkapan kamera kedua.	59
3.55	Citra hasil proses <i>plotting</i>	60
3.56	Citra hasil proses transformasi citra.	61
3.57	Citra hasil proses <i>plotting</i> dan penentuan area.	61
3.58	Tangkap citra pelanggar kecepatan	61
3.59	Tampilan utama <i>user interface</i> pemantauan pelanggaran kecepatan.	62
3.60	Tampilan utama pada aplikasi web.	63
4.1	Hasil pengujian perhitungan kecepatan dengan 30 <i>fps</i>	67
4.2	Percobaan Pelanggaran Kecepatan dengan 25 <i>fps</i>	68
4.3	Percobaan Pelanggaran Kecepatan.	69
4.4	Galat hasil pengujian.	71
4.5	Hasil pengujian jenis kendaraan.	72
4.6	Macam-macam Region of Interest	73
4.7	Data galat hasil pengujian dengan perbedaan pemilihan ROI.	74
4.8	Percobaan Pelanggaran Kecepatan.	77
4.9	Percobaan Pelanggaran Kecepatan.	78
4.10	Grafik regresi polinomial orde 5.	81

Halaman ini sengaja dikosongkan

DAFTAR TABEL

3.1	Spesifikasi <i>IP Camera</i>	29
3.2	Spesifikasi <i>IP Camera</i>	29
4.1	Hasil pengujian perhitungan kecepatan kendaraan dengan <i>frame rate</i> 30 <i>fps</i>	66
4.2	Hasil pengujian perhitungan kecepatan kendaraan dengan <i>frame rate</i> 25 <i>fps</i>	68
4.3	Hasil pengujian perhitungan kecepatan kendaraan dengan <i>frame rate</i> 20 <i>fps</i>	69
4.4	Hasil pengujian perhitungan kecepatan kendaraan dengan perbedaan kategori bayangan	70
4.5	Hasil pengujian jenis kendaraan	71
4.6	Hasil pengujian dengan perbedaan pemilihan ROI.	73
4.7	Hasil pengujian pelanggaran kecepatan	75
4.8	Percobaan Pelanggaran Kecepatan.	79
4.9	Pengujian durasi pengiriman data pelanggaran kecepatan ke database	80
4.10	Pengujian kecepatan dengan pendekatan regresi polinomial orde 5	82

Halaman ini sengaja dikosongkan

NOMENKLATUR

V	: Kecepatan
D	: Jarak tempuh
T	: Waktu tempuh
fps	: <i>frame per second</i>
D_{ij}	: Tingkat perbedaan (<i>dissimilarity degree</i>)
n	: Jumlah vektor
q_{ik}	: Vektor masukan
p_{jj}	: Vektor pembanding
p'	: Posisi piksel hasil transformasi
H	: <i>Homography Matrix</i>
p	: Posisi piksel pada gambar original
x'	: Posisi piksel hasil transformasi pada sumbu x
y'	: Posisi piksel hasil transformasi pada sumbu y
w	: Posisi piksel hasil transformasi pada sumbu z
x	: Posisi piksel pada gambar original di sumbu x
y	: Posisi piksel pada gambar original di sumbu y
$B_t(x, y)$: Citra latar pada waktu t
$F_t(x, y)$: Citra saat ini pada waktu t
α	: Kecepatan perbaharuan
$P[F(t)]$: Citra hasil proses <i>background subtraction</i>
$P[I(t)]$: Citra masukan
$P[B]$: Citra latar
$dst(x, y)$: Nilai piksel pada citra hasil
$src(x, y)$: Nilai piksel pada citra masukan
\bar{G}	: Nilai galat

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Kondisi lalu lintas pada setiap harinya semakin padat dengan kendaraan sehingga mempengaruhi akses dalam berkendara. Hal tersebut di perparah oleh pengendara yang kerap melanggar rambu - rambu lalu lintas, yang berakibat terjadinya kecelakaan lalu lintas. Berdasarkan data dari kepolisian bahwa jumlah korban kecelakaan lalu lintas pada tahun 2016 mencapai 105 ribu jiwa dengan jumlah korban meninggal sebanyak 25 ribu jiwa. Untuk menurunkan angka kecelakaan tersebut, maka dilaksanakannya beberapa program pembatasan kecepatan yaitu Program Zoss (Zona Selamat Sekolah), Program RASS (Rute Aman Selamat Sekolah), dan Program LRK (Lokasi Rawan Kecelakaan)[1].

Hartono, Pribadi [2] telah mengimplementasikan *image processing* untuk menghitung kecepatan kendaraan menggunakan *Traffic IP-Camera* dengan metode *Foreground Extraction*. Dalam penelitiannya, didapatkan tingkat akurasi tertinggi hingga 98 % pada kecepatan acuan 50 km/jam dan tingkat akurasi terendah hingga 93 % pada kecepatan 80km/jam. Hasil tersebut diperoleh dalam pengujian sistem dengan ukuran resolusi citra 720x576 piksel dengan jumlah citra perdetik sebanyak 25 fps.

Dalam mengimplementasikan *image processing* untuk menghitung nilai kecepatan kendaraan dibutuhkan beberapa metode agar mendapatkan hasil perhitungan kecepatan dengan tingkat akurasi yang tinggi. Dalam pemantauan pelanggaran kecepatan, selain mendeteksi kecepatan kendaraan, juga diperlukannya tangkapan citra pengendara yang melanggar kecepatan dengan jelas. Karena pemasangan *IP-Camera* membutuhkan infrastruktur yang tetap maka cakupan pengambilan citra lalu lintas juga terbatas, sehingga pengendara yang melanggar kecepatan hanya dapat diambil citranya

dari posisi atas. Penempatan posisi plat nomer yang beragam pada tiap kendaraan mengakibatkan penangkapan citra dari posisi atas kurang bisa maksimal. Sehingga dibutuhkan kamera lain dengan penempatan posisi yang berbeda. Penempatan posisi kamera 2 ini ditempatkan di simpang jalan dengan posisi lebih rendah dari pada posisi dari kamera 1. Cakupan citra yang dapat diambil oleh kamera 2 yaitu citra depan kendaraan. Sehingga data citra dari pengendara yang melanggar kecepatan lebih detail.

Dalam tugas akhir ini dikembangkan suatu sistem untuk memantau pelanggaran kecepatan kendaraan dengan memanfaatkan citra yang di ambil dari dua buah *IP-Camera*.

1.2 Permasalahan

Ketika terjadi pelanggaran, maka dibutuhkan suatu bukti digital berupa citra dari pelaku pelanggaran. Sistem pemantauan pelanggaran kecepatan saat ini masih dibutuhkan petugas untuk memantau dan mengambil citra pelanggar kecepatan. Jika citra yang ditangkap terdapat lebih dari satu pengendara, sistem ini belum bisa membedakan mana yang pelanggar dan mana yang bukan, jadi masih dibutuhkan petugas untuk membedakan secara manual.

1.3 Tujuan

Tujuan tugas akhir ini yaitu mengembangkan suatu sistem untuk memantau pelanggaran kecepatan kendaraan dengan memanfaatkan dua kamera yang harapannya bisa mendapatkan data citra lebih jelas dari kendaraan yang melanggar kecepatan.

1.4 Batasan masalah

Batasan masalah yang timbul dari permasalahan Tugas Akhir ini adalah :

1. Kondisi cuaca saat pengujian yaitu pada pagi, siang, dan sore hari.
2. Kondisi jalan saat pengujian yaitu lancar atau tidak macet.
3. Hanya mengambil citra (*capture*) pada kendaraan yang melanggar kecepatan, tidak sampai mengklarifikasi kendaraan dan plat nomer.

4. Data hasil *capture* disimpan di tempat penyimpanan dengan jaringan lokal.

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.

2. BAB II Dasar Teori

Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *image alignment*, *euclidean distance*, pengukuran kecepatan, *background reconstruction* dan lainnya.

3. BAB III Perancangan Sistem dan Implementasi

Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem dalam penelitian ini dan menganalisa sistem. Spesifikasi perangkat keras dan perangkat lunak yang diuji juga disebutkan dalam bab ini. Sehingga ketika akan dikembangkan lebih jauh, spesifikasi perlengkapannya bisa dipenuhi dengan mudah tanpa harus melakukan ujicoba perangkat lunak maupun perangkat keras lagi.

5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Sar-

an dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

Penelitian mengenai pemrosesan citra merupakan bidang penelitian yang sangat dibutuhkan dan dipakai dalam pemenuhan kebutuhan komputasi yang semakin beragam saat ini. Dalam tugas akhir ini, dilakukan penelitian mengenai pengembangan sistem pemantauan pelanggaran kecepatan menggunakan dua sensor kamera.

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Pengukuran Kecepatan

Hasil kecepatan objek bergerak dapat dihitung dengan mencari total jarak yang ditempuh dari awal objek bergerak sampai akhir objek bergerak dibagi dengan waktu yang ditempuh dalam melakukan perpindahan. perhitungan kecepatan tersebut dapat diketahui dengan menggunakan persamaan 2.1.

$$V = \frac{D}{T} \quad (2.1)$$

Pada persamaan 2.1, dimana V merupakan *velocity* atau kecepatan yang dicapai dalam melakukan perpindahan, D merupakan *distance* atau jarak yang ditempuh, dan T merupakan *time* atau waktu yang ditempuh dalam melakukan perpindahan.

Berdasarkan persamaan kecepatan 2.1, agar mendapatkan hasil perhitungan kecepatan laju kendaraan dari hasil rekaman video, nilai jarak didapatkan berdasarkan jarak referensi yang ditempuh dari awal objek bergerak sampai akhir objek bergerak yang telah ditentukan dan untuk waktu dapat diperoleh dari jumlah *frame* yang didapat saat objek melakukan perpindahan pada lingkup sesuai batas yang ditentukan. Sehingga proses perhitungan kecepatan dapat menggunakan persamaan 2.2.

$$Kecepatan = \frac{Panjang\ Garis \times fps \times 3600}{Jumlah\ Frame \times 1000} km/h \quad (2.2)$$

Pada persamaan 2.2, dimana panjang garis merupakan jarak yang sudah ditentukan dalam penelitian, *fps* merupakan *frame per second* yaitu kemampuan kamera dalam mengambil citra tiap detik

nya, dan jumlah *frame* merupakan banyak *frame* yang ditempuh oleh objek dalam batas garis yang sudah ditentukan [3].

2.2 IP Camera

IP Camera adalah CCTV (*Closed-circuit television*) yang menggunakan Protokol Internet untuk mengirimkan data gambar dan sinyal kendali atas *Fast Ethernet link*. Sejumlah *IP Camera* biasanya ditempatkan bersama-sama dengan perekam video *digital* (DVR) atau jaringan perekam *video* untuk membentuk sistem pengawasan video [4].

2.3 Euclidean Distance

Jarak digunakan untuk menentukan tingkat kesamaan (*similarity degree*) atau ketidaksamaan (*disimilarity degree*) dua vektor fitur. Tingkat kesamaan berupa suatu nilai dan berdasarkan nilai dua vektor fitur akan dikatakan sama atau tidak. (*Euclidean distance*) merupakan salah satu metode yang sering digunakan untuk menghitung kesamaan dari dua vektor. *Euclidean distance* menghitung akar dari kuadrat perbedaan dua vektor (*root of square difference between two vectors*). Perhitungan *Euclidean distance* dapat diketahui dengan menggunakan persamaan 2.3 [5].

$$D_{ij} = \sqrt{\sum_{k=1}^n (q_{ik} - p_{jk})^2} \quad (2.3)$$

Pada persamaan 2.3, dimana D_{ij} merupakan tingkat perbedaan n merupakan jumlah vektor, q_{ik} merupakan vektor masukan, dan p_{jk} merupakan vektor pembanding.

2.4 Projective Transformation

Projection merupakan metode yang sering digunakan untuk memetakan sebuah citra pada bidang citra lain. Ada banyak jenis dari *projection* seperti *Projection Transformation*, *Cylindrical Projection*, *Spherical Projection* dan lain lain. *Projective Transformation* juga dikenal sebagai *Perspective Transformation* atau *Homography*, yang di operasikan pada *homogeneous coordinat*. Meto-

de ini dilakukan dengan melakukan transformasi pada sebuah citra sehingga terproyeksikan pada bidang citra lain. Transformasi dilakukan dengan mengkalikan piksel - piksel pada citra yang akan diproyeksikan dengan *Homography Matrix*. *Homography Matrix* dapat di representasikan menggunakan persamaan 2.4. Berdasarkan persamaan 2.4, p' merupakan posisi piksel hasil transformasi, H merupakan *Homography Matrix* yang berupa matrix 3 x 3, dan p merupakan posisi piksel - piksel pada citra original. Setiap variabel pada persamaan 2.4 memiliki data seperti pada persamaan 2.5, dimana x dan y merupakan lokasi piksel pada citra masukan, sedangkan x' dan y' merupakan lokasi piksel pada citra hasil.

$$p' = Hp \quad (2.4)$$

$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.5)$$

Nilai pada variabel x' dan y' dapat ditemukan dengan melakukan normalisasi perkalian matrix menggunakan persamaan 2.6 dan persamaan 2.7 [6].

$$x' = \frac{h_{00} \times x + h_{01} \times y + h_{02}}{h_{20} \times x + h_{21} \times y + h_{22}} \quad (2.6)$$

$$y' = \frac{h_{01} \times x + h_{11} \times y + h_{12}}{h_{20} \times x + h_{21} \times y + h_{22}} \quad (2.7)$$

Dalam memperkirakan nilai dari matrix H maka persamaan 2.6 dan 2.7 dapat tulis menjadi persamaan yang berbeda agar mendapatkan persamaan yang homogen.

$$x'(h_{20}x + h_{21}y + h_{22}) = h_{00}x + h_{01}y + h_{02} \quad (2.8)$$

$$y'(h_{20}x + h_{21}y + h_{22}) = h_{01}x + h_{11}y + h_{12} \quad (2.9)$$

Berdasarkan persamaan 2.8 dan persamaan 2.9 terlihat bahwa koefisien dari h tampak linier, sehingga dari persamaan 2.8 dan 2.9 dapat di bangun ulang menjadi persamaan linier seperti pada persamaan 2.10 dan persamaan 2.11.

$$c_x h = 0 \quad (2.10)$$

$$c_y h = 0 \quad (2.11)$$

Dengan,

$$\begin{aligned} h &= (h_{00}, h_{01}, h_{02}, h_{10}, h_{11}, h_{12}, h_{20}, h_{21}, h_{22}) \\ c_x &= (x, y, 1, 0, 0, 0, -x'x, -x'y, -x') \\ c_y &= (0, 0, 0, x, y, 1, -y'x, -y'y, -y') \end{aligned}$$

Berdasarkan persamaan 2.10 dan 2.11, maka dapat dibentuk menjadi sistem persamaan linear seperti pada persamaan 2.12.

$$Ch = 0 \quad (2.12)$$

Dengan,

$$C = (c_{x1} c_{y1} \dots c_{xN} c_{yN})$$

Dari persamaan 2.12 dapat dibangun menjadi sebuah persamaan dalam matrik seperti pada persamaan 2.13. Pada persamaan 2.13 merupakan contoh bentuk matrik dengan titik koordinat masukan sebanyak empat titik koordinat.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2 x_2 & -x'_2 y_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2 x_2 & -y'_2 y_2 & -y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3 x_3 & -x'_3 y_3 & -x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3 x_3 & -y'_3 y_3 & -y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4 x_4 & -x'_4 y_4 & -x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4 x_4 & -y'_4 y_4 & -y'_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.13)$$

Dengan,

h_{00}	= faktor skala tetap dalam arah X dengan skala Y tidak berubah
h_{01}	= faktor skala dalam arah X sebanding dengan jarak Y dari asal
h_{02}	= Translasi asal dalam arah X
h_{10}	= faktor skala dalam arah Y sebanding dengan jarak X dari asal
h_{11}	= Faktor skala tetap dalam arah Y dengan skala X tidak berubah
h_{12}	= Translasi asal dalam arah Y
h_{20}	= faktor skala proporsional X dan Y dalam fungsi X
h_{21}	= faktor skala proporsional X dan Y dalam fungsi Y

Berdasarkan persamaan 2.12 dimana merupakan normalisasi dari persamaan sebelumnya hingga menjadi persamaan linier. Nilai matrik h memiliki nilai yang sama dengan *Eigenvector* dari $C^T C$ dengan syarat nilai *Eigenvalue* bernilai 0 atau mendekati 0 [7]. C merupakan matrik berordo 9×9 seperti pada persamaan 2.13 dan T merupakan *Transposed matrix*.

Berdasarkan keterangan pada setiap elemen matrik h di persamaan 2.13, menunjukan bahwa *projective transformation* itu meliputi beberapa transformasi seperti translation, rotation, shearing, dan scaling [8].

2.5 Background Reconstruction

Background Reconstruction atau pembuatan citra latar adalah suatu metode pengolahan citra digital untuk membentuk citra latar belakang dari rangkaian *frame* atau video. Proses *Background Reconstruction* ini di tujukan agar mendapatkan model citra latar yang pada umumnya digunakan dalam proses *Background Subtraction* sebagai pembanding antara citra latar dengan citra yang sekarang. terdapat banyak metode untuk merekontruksi citra latar dan hingga saat ini masih terus dkembangkan agar mendapatkan model citra yang optimal [9].

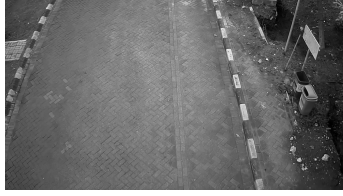
Metode yang digunakan dalam proses rekontruksi citra latar menggunakan fungsi *Accumulate Weight*, dimana membutuhkan dua citra untuk diproses. Formula *Accumulate Weight* dapat dilihat pada persamaan 2.14.

$$B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha F_t(x, y) \quad (2.14)$$

Bedasarkan persamaan 2.14, b_t merupakan citra latar pada waktu dan α merupakan *updatating rate*, pembentukan citra latar dilakukan dengan menghitung jumlah bobot nilai piksel dari citra masukan kemudian dilakukan perhitungan rata-rata dari tiap piksel perframe atau citra masukan dengan koordinat yang sama [10] yang di ilustrasikan pada gambar 2.1. Perhitungan rata-rata dilakukan secara berjalan atau kontinu (*Running average*) sampai batas jumlah citra masukan yang dikehendaki. Hasil dari proses pembuatan citra latar diilustrasikan pada gambar 2.2.



Gambar 2.1: Citra masukan dari *background reconstruction*.



Gambar 2.2: Hasil proses *background reconstruction*.

2.6 Background Subtraction

Background Subtraction atau Subtraksi Citra Latar adalah sebuah teknik pada pengolahan citra digital untuk mengekstrak citra *foreground*. Pada proses ini diperlukan dua citra masukan, yaitu citra sekarang dan citra latar. Pendekatan yang dilakukan digunakan untuk pendeteksian objek bergerak dengan membandingkan citra sekarang dengan citra latar hasil proses *background reconstruction*. Tujuan dari proses *background subtraction* adalah untuk mendapatkan citra *foreground*. Persamaan *frame difference* dalam *background subtraction* dapat dilihat pada persamaan 2.15 [11].

$$P[F(t)] = P[I(t)] - P[B] \quad (2.15)$$

Pada persamaan 2.15, digunakan untuk menentukan citra *foreground*, dimana $P[F(t)]$ merupakan citra hasil subtraksi antara citra masukan $P[I(t)]$ terhadap citra latar $P[B]$ pada waktu t [9]. Pada Gambar 2.3.b mengilustrasikan citra hasil *background subtraction*, sedangkan pada gambar 2.3.a mengilustrasikan citra masukan.



(a) citra *grayscale*.



(b) Citra *foreground*.

Gambar 2.3: Citra sebelum dan sesudah proses *Background subtraction*.

2.7 Thresholding

Pada pemrosesan citra digital, *Thresholding* atau subtraksi citra biner adalah suatu teknik untuk segmentasi citra digital. *Thresholding* membutuhkan citra masukan dan nilai *thresh* agar menghasilkan citra keluaran. Citra masukan yang digunakan yaitu citra abu-abu. Proses *thresholding* yaitu dengan cara membandingkan nilai pixel (x,y) pada citra masukan dengan nilai *thresh*. Jika nilai pixel (x,y) pada citra masukan lebih besar dari pada nilai ambang, maka nilai pixel (x,y) tersebut pada citra diganti ke suatu nilai [11]. Operasi *thresholding* dapat dinyatakan dengan persamaan 2.16.

$$dst(x,y) = \begin{cases} maxVal & \text{if } src(x,y) > thresh \\ 0 & \text{otherwise} \end{cases} \quad (2.16)$$

Pada persamaan 2.16 merupakan salah satu jenis dari bentuk *thresholding* yaitu *binary thresholding*. Pada persamaan 2.16 diketahui bahwa *maxVal* adalah parameter yang dijadikan nilai pixel (x,y) pengganti dari nilai pixel (x,y) pada citra masukan. pada *binary thresholding*, *maxVal* bernilai 255 atau pixel warna putih. Setiap pixel (x,y) pada citra masukan *src* dikomparasi dengan nilai *thresh*. Jika nilai pixel (x,y) pada citra *src* lebih besar dari pada nilai *thresh*, maka nilai pixel *dst* (x,y) diubah menjadi nilai *thresh* atau 255. Sedangkan Jika nilai pixel (x,y) pada citra *src* tidak lebih besar dari pada nilai *thresh*, maka nilai pixel *dst* (x,y) diubah menjadi nilai nol atau citra warna hitam. Hasil dari proses *thresholding* dapat diilustrasikan pada gambar 2.4.



(a) Citra hasil subtraksi.

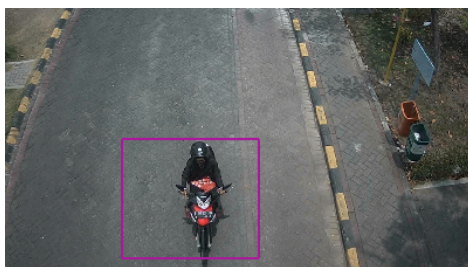


(b) Citra hasil *thresholding*.

Gambar 2.4: Citra sebelum dan sesudah proses *binary thresholding*.

2.8 Image Cropping

Image cropping atau pemotongan citra adalah cara memotong sebagian daerah pada sebuah citra untuk membuang bagian yang tidak diinginkan [12]. Proses pemotongan citra menggunakan fungsi *Rect* (*int x*, *int y*, *int width*, *int height*) agar menghasilkan potongan citra dalam bentuk segi empat dan berdasarkan lokasi yang dihendaki. Parameter (*int x*, *int y*) pada fungsi *Rect()* menunjukan lokasi pixel *x,y* pada citra masukan yang akan *cropping* dan parameter (*int width*, *int height*) menunjukan cakupan citra yang akan di *cropping* [13]. Pada Gambar 2.5.b mengilustrasikan citra hasil *cropping*, sedangkan pada gambar 2.5.a mengilustrasikan citra masukan berserta *rectangle*.



(a) Citra masukan berserta *Rectangle*.



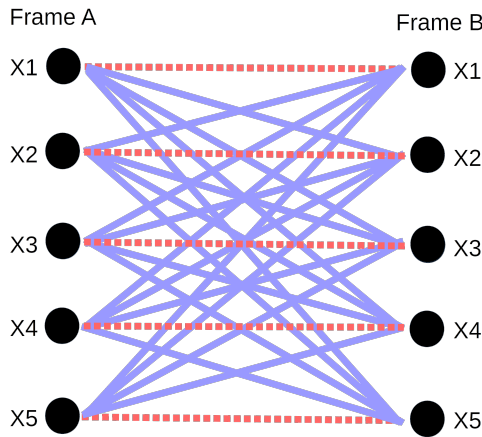
(b) Citra hasil *cropping*.

Gambar 2.5: Citra masukan dan citra keluaran proses *image cropping*.

2.9 Algoritma *Matching Based*

Matching-based merupakan algoritma untuk mencari hubungan antar *frame* yang berdasarkan nilai kesamaan terbesar. Misalkan dari *frame A* terdapat beberapa titik koordinat yaitu $X_a = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ dan dalam *frame B* terdapat beberapa titik yaitu $X_b = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, maka titik dari setiap *frame* dicari kecocokan atau kesesuaiannya. Nilai kesesuaian atau kesamaan antara dua titik dapat menggunakan nilai hasil perhitungan *Euclidean distance* terkecil [5]. *Euclidean distance* adalah persamaan yang menghitung jarak antara dua titik.

Pada gambar 2.6 merupakan ilustrasi algoritma *matching-based*, dimana setiap titik pada *frame A* membandingkan dengan setiap titik pada *frame B*. Pada gambar 2.6 terlihat bahwa setiap titik yang terhubung dengan garis biru dan garis merah merupakan hubungan yang telah terjadi antar tiap titik pada setiap frame. Garis biru menunjukkan hubungan yang tidak memiliki kesesuaian antar dua titik, sedangkan garis merah menunjukkan kesesuaian atau kesamaan terbesar yang diperoleh oleh dua titik [14].



Gambar 2.6: *complete bi-partite graph.*

2.10 Penelitian Terkait Perhitungan Jumlah Kendaraan dan Deteksi Kecepatan Kendaraan

Berkaitan dengan topik tugas akhir ini, terdapat beberapa penelitian mengenai penghitungan jumlah kendaraan dan perhitungan kecepatan kendaraan yang diaplikasikan untuk Intelligent Transportation System (ITS). Tiga diantaranya sebagai berikut.

1. Penelitian yang berjudul "*Real Time Vehicle Counter System for Intelligent Transportation System*" [15] menggunakan *Traffic IP-Camera* untuk mendapatkan citra masukan yang diproses secara *realtime*. Proses untuk penghitungan jumlah kendaraan dibagi menjadi beberapa prosedur yaitu ekstraksi citra

latar, segmentasi objek, pelabelan kendaraan dan klasifikasi objek untuk membedakan kendaraan roda dua dan kendaraan roda empat. Hasil perhitungan jumlah kendaraan untuk kondisi tanpa adanya bayangan atau objek yang tidak diinginkan, sistem menghasilkan kesuksesan pencocokan hingga 97 % untuk kendaraan roda dua dan 95 % untuk kendaraan roda empat.

2. Penelitian yang berjudul "Traffic IP Camera untuk Menghitung Kendaraan Roda Empat Menggunakan Metode Luasan Piksel" [9] menggunakan *Traffic IP-Camera* sebagai alat untuk mendapatkan citra masukkan dan diproses menggunakan *Single Board Computer*. Metode luasan piksel digunakan untuk mendeteksi kendaraan. Berdasarkan pengujian secara langsung di tempat, sistem menghasilkan perhitungan jumlah kendaraan roda empat dengan akurasi hingga 82,18 % pada siang hari dan 88,30 % pada malam hari.
3. Penelitian yang berjudul "Sistem Deteksi Kecepatan Kendaraan Bermotor pada *Real Time Traffic Information System*" [2] menggunakan metode *background reconstruction* untuk membuat citra *background* dan *frame difference* untuk mendapatkan citra *foreground*. Perubahan perpindahan objek pada suatu jarak tertentu dengan waktu tertentu digunakan sebagai referensi untuk mendapatkan perhitungan kecepatan. Ketepatan pengukuran mencapai 98 % pada kecepatan acuan 50 kmj. Akurasi terendah terjadi pada kecepatan kendaraan 80 kmj dengan ketepatan hanya 93 %. Tingkat akurasi tergantung pada besarnya pergeseran per citra adanya bayangan dan terjadinya penumpukan obyek kendaraan.

Halaman ini sengaja dikosongkan

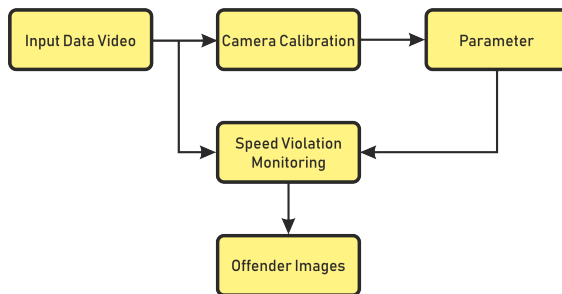
BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan.

3.1 Desain Sistem

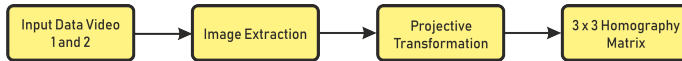
Tugas akhir ini merupakan salah satu bentuk dari pengolahan citra digital yang tujuannya untuk memantau pelanggaran kecepatan kendaraan yang lewat pada suatu jalan. Sistem perangkat lunak ini dibangun pada komputer personal dengan citra masukan dari dua sudut pandang yang ditangkap menggunakan dua kamera IP. Proses kerja dari sistem ditunjukkan pada pada Gambar 3.1.



Gambar 3.1: Diagram alir sistem pemantauan pelanggaran kecepatan.

Berdasarkan diagram alir pada gambar 3.1, setelah data video dimasukan maka terdapat beberapa proses yang berlangsung yaitu kalibrasi kamera yang menghasilkan sebuah parameter. Parameter yang didapat yaitu sebuah matrix 3x3. Hasil parameter yang didapat digunakan untuk proses perhitungan kecepatan kendaraan dan proses *plotting point*, dimana keluaran yang didapat adalah citra pelanggar kecepatan berkendara.

Pada gambar 3.2 menunjukan diagram alir metodologi untuk proses kalibrasi kamera. Inisialisasi data video merupakan proses yang dilakukan pertama kali. setiap data masukan video dari masing masing *IP camera* memiliki sudut pandang yang berbeda. Kedua video masukan tersebut kemudian di ekstraksi menjadi satu *frame* untuk tiap data masukan video.

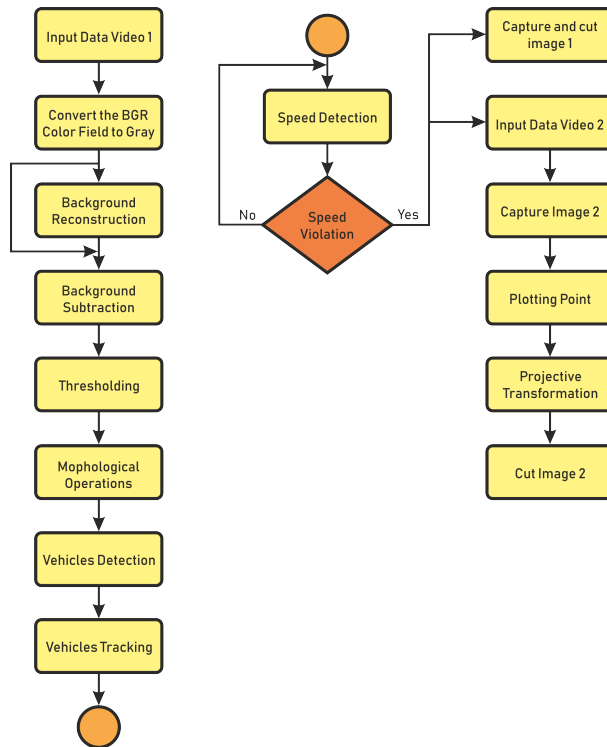


Gambar 3.2: Diagram alir metodologi penelitian untuk kalibrasi kamera.

Frame yang didapat dari dua sudut pandang yang berbeda, setiap *frame* di transformasikan ke bidang yang sama sesuai kondisi nyata guna membantu dalam pengukuran kecepatan kendaraan dan juga sebagai kalibrasi antara citra 1 dan citra 2. Hasil dari proses transformasi mendapatkan homografi matrix 3×3 untuk tiap *frame* dari citra 1 dan citra 2.

Pada gambar 3.3 merupakan diagram alir metodologi proses pemantauan pelanggaran kecepatan, dimana data masukan video yang diolah dalam proses pendeteksian kecepatan adalah data masukan video 1. Dalam mendeteksi sebuah objek diperlukannya citra latar dan citra sekarang untuk di bandingkan. Proses pembuatan citra latar dilakukan dengan cara merekonstruksi citra atau *background reconstruction*. Hasil dari *background reconstruction* menjadi data masukan untuk proses *background subtraction*, dimana hasil dari *background subtraction* adalah citra *foreground*.

Citra *foreground* yang sudah didapat, selanjutnya dilakukan proses *thresholding*. Proses ini berfungsi untuk mengubah nilai pixel keabu-abuan menjadi nilai hitam (0) atau putih (1) berdasarkan nilai ambang dari *threshold*. Setelah dilakukan proses *thresholding* maka ditemukan *noise* atau pixel yang tidak diinginkan, sehingga diperlukan pembersihan citra dengan metode erosi untuk mengurangi *noise* dan dilasi untuk mempertegas objek tujuan pada citra. Tujuan dari pembersihan citra atau operasi morfologi adalah untuk memperjelas bentuk objek yang dideteksi.



Gambar 3.3: Diagram alir metodologi penelitian untuk pemantauan pelanggaran kecepatan.

Setelah mendapatkan kumpulan piksel warna putih hasil dari proses sebelumnya kemudian mencari kontur untuk menentukan area yang diduga sebagai kendaraan. Area yang telah terpasang kontur dapat diketahui posisi koordinat area tersebut menggunakan fungsi *moment*. Penggunaan *moment* pada setiap kontur yang terdeteksi dapat mengetahui titik pusat area yang terpasang kontur. Titik pusat merupakan parameter yang baik untuk mengetahui lokasi setiap kendaraan yang terdeteksi.

Banyak nya kendaraan yang terdeteksi, perlu adanya penjejak-an objek untuk membedakan setiap kendaraan yang terdeteksi agar

dapat menghitung nilai kecepatan tiap kendaraan. Penjejak objek dapat menggunakan pendetakan perhitungan jarak. Perhitungan jarak menggunakan persamaan *Euclidean Distance*. Titik pusat dari setiap objek yang terdeteksi menjadi data dalam perhitungan *Euclidean Distance*, dimana jika hasil perhitungan nya memiliki nilai yang hampir sama tiap *frame* nya, maka objek itu didefinisikan dalam satu objek yang sama.

Setelah dapat membedakan setiap kendaraan yang terdeteksi, maka dilakukannya perhitungan kecepatan tiap kendaraan tersebut. Perhitungan kecepatan menggunakan konsep dari persamaan fisika dasar dalam menentukan kecepatan suatu objek. Nilai jarak didapatkan dari pengukuran dikondisi nyata. Untuk mendapatkan titik koordinat objek di kondisi nyata berdasarkan titik koordinat pada citra, maka diperlukannya transformasi titik koordinat dari citra yang ditangkap ke kondisi nyata. Pada proses sebelumnya yaitu kalibrasi kamera adalah proses mentransformasikan titik koordinat sebuah citra ke kondisi nyata, dimana pada proses tersebut menghasilkan sebuah homografi matrix H . Matrix H yang didapat, dikalikan dengan setiap titik pusat pada objek yang terdeteksi. Dalam proses pengukuran jarak diberikan sebuah penanda. Penanda yang tertangkap oleh kamera akan dijadikan sebagai acuan untuk titik awal dan titik akhir yang harus dilalui objek yang terdeteksi. Banyaknya *frame* yang ditempuh objek tersebut antara titik acuan awal dengan akhir, dapat dijadikan sebagai waktu tempuh sebuah objek yang bergerak. Kualitas tangkap kamera atau fps (*frame per second*) pun berpengaruh dalam perhitungan kecepatan.

Jika hasil dari proses perhitungan kecepatan menunjukkan bahwa objek tersebut melaju diatas ambang batas kecepatan, maka citra kendaraan tersebut akan tertangkap oleh kamera pertama dan disaat yang hampir bersamaan, kamera 2 juga menangkap satu citra di frame tersebut. Citra yang didapat oleh kamera 2 akan dilakukan proses *plotting point* untuk menentukan titik lokasi kendaraan. Kendaraan pada citra yang ditangkap oleh kamera 2, terlihat miring atau tidak lurus dari posisi depan. Sehingga citra plat nomer kendaraan tersebut juga terlihat miring. Dalam kondisi tersebut maka dilakukannya proses *projective transformation* agar citra plat nomer kendaraan tersebut terlihat lurus dari posisi depan. Diakhir proses

dilakukan pemotongan citra, yang hasilnya yaitu berupa citra dari kendaraan yang melanggar kecepatan.

3.2 Work Flow

Pengerjaan tugas akhir ini dibagi menjadi beberapa tahapan berdasarkan metodologi penelitian, yaitu :

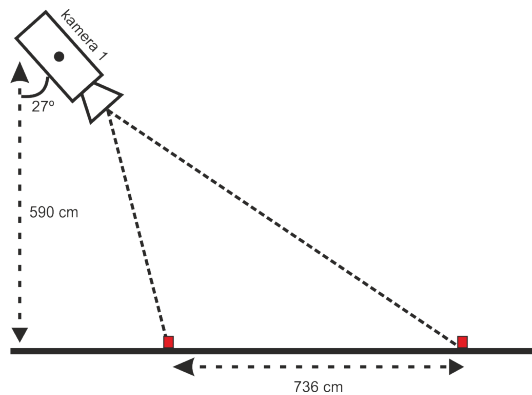
1. Akusisi Data.
2. Kalibrasi Kedua Kamera.
3. Penjejukan Kendaraan.
4. Perhitungan Kecepatan Kendaraan.
5. Penentuan Pelanggaran Kecepatan.

3.3 Akuisisi Data

Pada tahap akuisisi data video, terdapat dua data video yang di proses dalam penelitian ini yaitu data video yang di tangkap dari posisi atas dan data video yang ditangkap dari posisi depan objek.

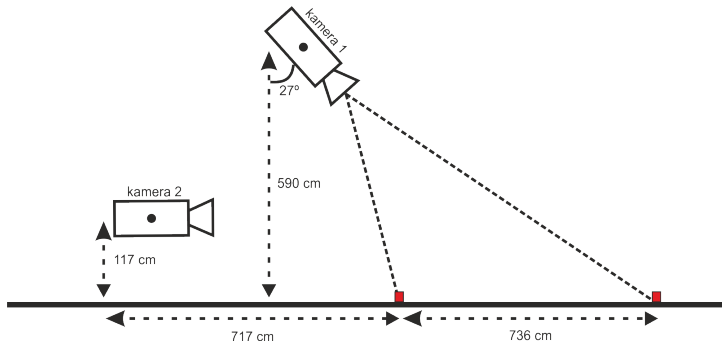
3.3.1 Persiapan dan Peletakan Kamera

Akuisisi data video pada penelitian ini menggunakan dua *IP Camera* masing masing memiliki cakupan tangkap video yang berbeda. Berdasarkan pada gambar 3.4, kamera pertama diletakan pada sebuah tiang jalan dengan ketinggian 590 cm.



Gambar 3.4: Skema kamera pertama.

Peletakan *IP Camera* pada tiang jalan tersebut memiliki kemiringan sebesar 27 derajat dengan cakupan luasan jalan yang dapat ditangkap sebesar 583 x 736 cm seperti pada gambar 3.5.

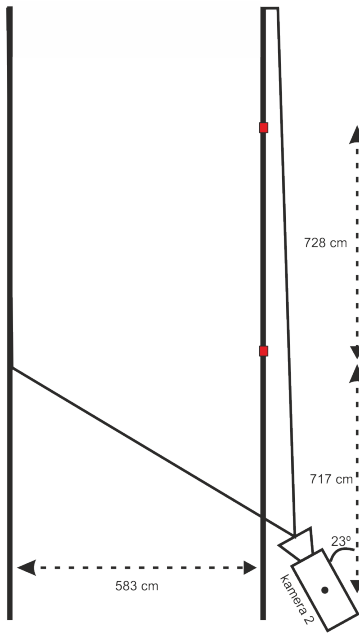


Gambar 3.5: Skema kedua kamera.

Berdasarkan pada gambar 3.6, kamera 2 diletakan di posisi bawah dengan jarak 717 cm dari jarak batas bawah cakupan kamera pertama. Kamera 2 di letakan pada sebuah penyangga kaki tiga atau tripod dengan ketinggian 117 cm dan sudut kemiringan kamera sebesar 23 derajat. Cakupan citra yang di tangkap oleh kamera 2 adalah objek pada posisi depan.

3.3.2 Pemasangan Penanda

Pemasangan penanda dilakukan untuk di proses pada tahap kalibrasi kamera. Berdasarkan gambar 3.7, terdapat enam penanda yang di pasang di bahu jalan. ukuran jarak antar penanda disesuaikan dengan lebar dan panjang jalan. Selain digunakan untuk kalibrasi kamera, pemasangan penanda ini juga di gunakan sebagai titik acuan untuk penentuan titik awal dan titik akhir dalam proses perhitungan kecepatan. Area yang sudah terpasang penanda akan di ambill citra dari area tersebut dari sudut pandang dari setiap kamera yang sudah terpasang. Citra area penanda yang sudah tertangkap oleh tiap kamera akan menunjukkan koordinat posisi tiap penanda pada citra 1 dan citra 2 berbeda, sehingga kedua citra tersebut yang akan di proses pada tahap kalibrasi kamera.



Gambar 3.6: Skema kamera 1.



Gambar 3.7: Pemasangan penanda pada pinggir jalan.

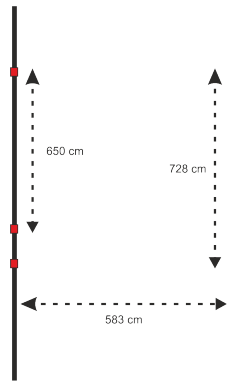
Sesuai dengan ketinggian penempatan kamera dan sudut kemiringan kamera, pada citra hasil tangkapan oleh kamera pertama bahwa pada citra tersebut menunjukkan bahwa Terdapat enam penanda yang tertangkap kamera pertama. jika dilihat oleh mata dari citra tersebut, terlihat bahwa posisi marker sebelah kiri dan kana agak berbeda sedangkan saat proses pemasangan penanda di kondisi nyata, posisi marker sebelah kiri dan kana hampir sama. dalam hal ini citra yang ditangkap oleh kamera pertama pelunya kalibrasi agar citra yang diproses menghasilnya data yang benar. Pada gambar 3.8 terlihat bahwa citra area penanda yang di tangkap oleh kamera 2. Cangkupan citra yang yang ditangkap oleh kamera 2 terlihat dapat menangkap objek dari posisi depan sesuai dengan peletakan kamera 2 dengan kemiringan dan tinggi penyangga yang sudah di atur sebelumnya.



Gambar 3.8: Pemasangan penanda pada pinggir jalan.

Posisi letak penanda pada citra yang ditangkap kamera 2, sangatlah tidak sama antara penanda disebelah kiri dan kanan. pada citra tersebut juga terlihat terdapat enam penanda yang tertangkap oleh kamera 2. tujuan dari peletakan penanda dimana, pada area penanda tersebut ditangkap oleh kedua kamera adalah untuk menentukan hubungan atau korelasi antara citra hasil tangkapan kamera pertama dengan citra hasil tangkapan kamera 2.

Peletakan penanda juga bertujuan untuk menentukan batas jarak dalam perhitungan kecepatan. Peletakan posisi penanda menunjukkan bahwa ada tiga baris dan dua kolom. Dalam perhitungan kecepatan, batas jarak yang digunakan yaitu antara baris pertama dan baris kedua, dimana pada baris awal menunjukkan titik acuan awal dan baris kedua menunjukkan titik acuan akhir. Pada gambar 3.9, jarak antara titik acuan awal dan titik acuan akhir sejauh 650 cm. Jarak antara baris pertama dan baris ketiga merupakan cakupan yang sanggup di tangkap oleh kamera pertama.



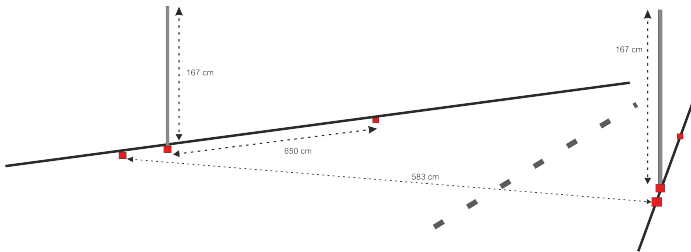
Gambar 3.9: Pemasangan penanda pada pinggir jalan.

Pemberian enam penanda yang tersebar di bahu jalan, hanya sebagai kalibrasi antara citra kamera pertama dengan kamera 2. dalam hal penelitian ini, penggunaan kamera 2 berfungsi sebagai fitur tambahan agar mendapatkan data citra pelanggar kecepatan dari posisi depan yang bertujuan untuk mendapatkan plat nomer kendaraan tersebut dengan jelas. Dikarenakan citra yang ditangkap oleh kamera 2 tidak bisa di letakan di tengah jalan karena infrastruktur yang tidak memadai, maka dari itu peletakan kamera 2 di posisi di bahu jalan. Hal tersebut membuat citra tangkapan dari kamera dua tidak bisa menangkap objek secara lurus dari depan. dalam hal tersebut citra tangkapan kamera 2 perlu di proses agar mendapatkan citra plat nomer dari kendaraan yang melanggar secara lurus dari depan. Sebelum citra yang ditangkap kamera 2 di

proses, diperlukannya sebuah penanda yang menunjukkan kemiringan dari citra yang ditangkap kamera 2. Dalam hal tersebut penanda yang digunakan adalah sesuatu yang panjang dan diletakan di kedua sisi bahu jalan seperti pada gambar 3.10 dan gambar 3.11.



Gambar 3.10: Pemasangan penanda pada pinggir jalan.



Gambar 3.11: Pemasangan penanda pada pinggir jalan.

Berdasarkan gambar 3.10 terlihat bahwa ada pemasangan penanda menggunakan sebuah pipa yang memiliki panjang 167 cm. Penentuan panjang pipa disesuaikan dengan tinggi dari lokasi plat nomor kendaraan. Pipa dengan tinggi tersebut dianggap cukup untuk mencakup citra dari sebuah plat nomor kendaraan yang melangar kecepatan.

Peletakan semua penanda hanya sebagai kalibrasi awal, dimana setiap titik lokasi dari tiap penanda akan di ambil dan di proses pada tahap selanjutnya. Pada penanda yang menggunakan pipa, akan diambil data titik ujung pipa bagian atas pada setiap peletakan pipa tersebut.

3.3.3 Lokasi Akuisisi Data

Pengambilan data video pada penelitian ini menggunakan dua kamera. Sesuai dengan infrastruktur yang ada yaitu sudah terpasangnya *IP Camera* pada tiang jalan di area lingkungan Kampus ITS, maka pada area tersebut yang menjadi lokasi dalam penelitian ini. Jalan pada area tersebut merupakan salah satu jalan utama di lingkungan kampus ITS yang sering di lewati kendaraan. Dengan kondisi jalan yang cukup panjang tak heran jika sering adanya pengendara yang melaju dengan kencang. Sesuai dengan topik penelitian ini terkait pemantauan pelanggaran kecepatan berkendara, maka area pada jalan tersebut sangat cocok untuk dilakukan penelitian ini. Pada gambar 3.12, lingkaran merah menunjukkan lokasi dilakukannya penelitian ini, dimana terletak pada jalan Teknik Mesin yang berdekatan dengan Departemen Teknik Elektro ITS.



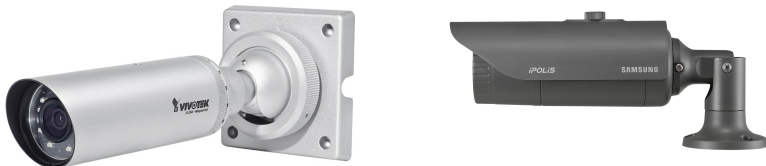
Gambar 3.12: Lokasi akuisisi data.

Pengambilan data video dilakukan pada waktu pagi, siang, dan sore hari. Kondisi cuaca seperti cerah berawan atau cerah tak berawan sangat berpengaruh dalam proses pengolahan data video

yang ditangkap. Terdapat banyak pohon yang tumbuh di area lingkungan kampus ITS, jika dalam kondisi cerah tak berawan maka akan banyak bayangan yang dihasilkan oleh pohon-pohon di area lingkungan ITS. Hal tersebut sangat mempengaruhi dalam pemrosesan data video yang ditangkap oleh kamera pertama.

3.3.4 IP Camera yang Digunakan

Berdasarkan gambar 3.13, terdapat dua *IP Camera* yang digunakan yaitu Vivotek IB8354-C dan Samsung Techwin SNO-5084, telah dikonfigurasi dengan pengaturan yang sama.



(a) Vivotek IB8354-C.

(b) Samsung Techwin SNO-5084R.

Gambar 3.13: Jenis kamera yang digunakan.

Dalam penelitian ini, konfigurasi jumlah *frame rate* yang digunakan pada setiap kamera yaitu 30 *fps* (*frame per second*) dengan resolusi HD (1280 x 720 piksel). Dalam penggunaan dua *IP Camera* ini, dibutuhkan suatu penghubung agar dapat diakses melalui satu *personal computer*. Dalam penelitian ini menggunakan switch sebagai penghubung antara kedua *IP Camera* ke *personal computer*. switch adalah perangkat jaringan komputer yang berfungsi sebagai konektor. Jenis switch yang digunakan adalah AT-GS900 dengan 8 *port*. Pada tabel 3.1 [16] dan 3.2 [17] merupakan spesifikasi dari *IP Camera* yang digunakan. Berdasarkan spesifikasi tersebut, dalam mengatur konfigurasi *IP Camera* harus menggunakan *browser* berdasarkan spesifikasi tiap *IP Camera*.

3.3.5 Pengambilan Data Video

Pengambilan data video dilakukan sesuai kebutuhan untuk pengujian. Terdapat tiga data video yang diambil berdasarkan waktu yang telah ditentukan yaitu pada pagi, siang, dan sore hari, dimana

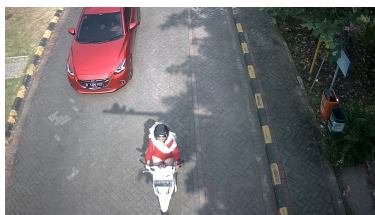
Tabel 3.1: Spesifikasi *IP Camera*.

Specifications IB8354-C	
System information	
CPU	Multimedia SoC (System-on-Chip)
Flash	128 MB
RAM	256 MB
Camera Features	
Image sensor	1/3" Progressive CMOS"
Maximum resolution	1280x1024 (1.3MP)
Lens type	Fixed-focal
Focal length	f = 4 mm
Shutter time	1/5 sec. to 1/32,000 sec
General connectors	RJ-45 for PoE connection
Video compression	H.264 & MJPEG
Maximum frame rate	30 fps @ 1280x1024
Sytem Requitments	
Operating system	Microsoft Windows 7/vista/XP/2000, MAC OS
Browser	Mozilla Firefox 7~10
	Internet Explorer 7/8/9/10

Tabel 3.2: Spesifikasi *IP Camera*.

Specifications SNO-5084R	
System information	
Scanning system	Progressive
Total pixels	1,312(H) x 1,069(V)
S/N Ratio	50dB
Camera Features	
Image sensor	1/3" 1.37M CMOS (IMX238)
Maximum resolution	1280x1024 (1.3MP)
Lens type	Fixed-focal
Focal length	3 ~8.5mm
Max. Aperture ratio	F1.2
General connectors	RJ-45 (10/100BASE-T)
Video compression	H.264 (MPEG-4 part 10/AVC), MJPEG
Maximum frame rate	60 fps @ all resolutions
Sytem Requitments	
Operating system	"Windows 8/7/vista/XP/2000, MAC OS X10.7
Browser	Mozilla Firefox 7~10
	Internet Explorer 7/8/9/10

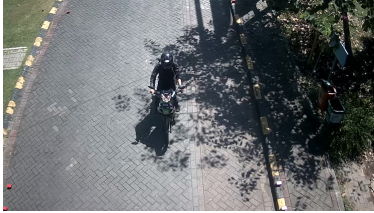
pada kondisi waktu tersebut akan dikategorikan data video tidak ada bayangan dan data video ada bayangan. Pengambilan data video untuk keperluan kalibrasi kamera hanya membutuhkan satu frame saja, dimana pada frame tersebut sudah terpasang penanda yang dibutuhkan. Setelah itu dilakukannya perekaman video untuk proses pengujian. Ukuran resolusi yang diajukan adalah 1280x720 piksel dengan kecepatan merekam sebesar 30 fps (*frame per second*). Pada gambar 3.14 terlihat bahwa data video diambil pada pagi hari pukul 08.45.



(a) Citra kamera pertama.

Gambar 3.14: Perekaman video pada pagi hari.

Pada gambar 3.15 merupakan data video yang diambil pada siang hari pukul 13.00, dimana pada gambar tersebut terlihat intensitas bayangan yang didapat cukup tinggi. Data video yang diambil pada siang dan pagi hari merupakan kategori data video ada bayangan. Pada gambar 3.16 merupakan data video yang diambil pada sore sekitar pukul 16.20. Pada data video tersebut terlihat tidak nampak bayangan dari objek sekitarnya. Hal tersebut merupakan kondisi yang cukup ideal untuk di proses pada tahap selanjutnya.



(a) Citra kamera 1



(b) Citra kamera 2

Gambar 3.15: Perekaman video pada siang hari



(a) Citra kamera 1



(b) Citra kamera 2

Gambar 3.16: Perekaman video pada sore hari.

3.4 Kalibrasi Kedua Kamera

Kalibrasi kedua kamera adalah proses untuk mengatur sebuah citra hasil tangkapan kamera pertama agar memiliki kolerasi dengan citra dua hasil tangkapan kamera 2. Pada proses kalibrasi ini menghasilkan sebuah parameter yaitu matrix 3×3 .

3.4.1 Ekstraksi Citra

Ekstraksi citra adalah dekode data video menjadi satuan citra. Video merupakan citra yang berjalan pada jumlah *fps* tertentu. Proses ekstraksi citra ini dilakukan setelah data masukan dimuat dalam program. Proses ekstraksi ini dilakukan agar mendapatkan satu frame dari tiap data video masukan. Terdapat dua data video masukan, maka pada proses ekstraksi ini menghasilkan dua citra dari dua data masukan video.

Pada Gambar 3.17 merupakan proses ekstraksi citra dari video masukan 1. Pada proses tersebut menghasilkan satu citra untuk video hasil tangkapan kamera 1.



Gambar 3.17: Proses ekstraksi citra 1.

Pada Gambar 3.18 merupakan proses ekstraksi citra dari video masukan 2. Pada proses tersebut juga menghasilkan satu citra untuk video hasil tangkapan kamera 2.



Gambar 3.18: Proses ekstraksi citra 2.

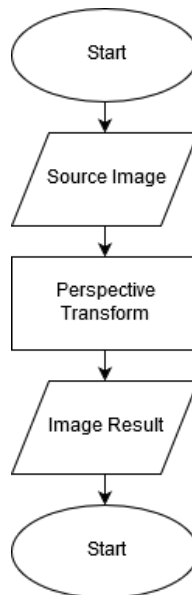
3.4.2 Projective Transformation

Pada tahap ini yaitu mentransformasikan citra yang di dapat dari proses ekstraksi ke suatu bidang lain. Bidang lain yang dimaksud yaitu kondisi pengujian dalam ukuran sesungguhnya. Sebelum melakukan transformasi diperlukannya beberapa titik koordinat dari citra sumber (*image source*) dan titik koordinat tujuan [18]. Titik koordinat dari citra sumber didapatkan dengan cara memetakan secara manual setiap posisi penanda yang terdapat pada citra. Titik koordinat tujuan didapatkan dengan cara pengukuran posisi jarak

penanda secara nyata. Citra sumber yang digunakan pada tahap ini terdapat dua citra yang masing-masing ditangkap oleh tiap kamera, sehingga setiap citra sumber memiliki enam buah koordinat untuk lokasi penanda.

Pada proses transformasi proyektif ini yang digunakan sebagai kalibrasi antar kamera 1 dan kamera 2 membutuhkan titik koordinat citra sumber dari setiap tangkapan kamera dan titik koordinat dari hasil pengukuran di kondisi nyata. Setiap citra hasil proses ekstraksi di transformasikan kesuatu bidang nyata yang sama [19]. Proses transformasi dari setiap citra akan menghasilkan sebuah matrix 3×3 yang berbeda-beda.

Pada Gambar 3.19 merupakan diagram alir proses perspective transform, dimana *input* yang dimaksud adalah citra sumber beserta titik koordinat yang sudah dipetakan. *Output* dari proses ini yaitu sebuah citra yang sudah di transformasikan beserta sebuah matrix 3×3 .



Gambar 3.19: Diagram alir proses *perspective transform*.

Pada gambar 3.20 dan gambar 3.21 merupakan proses *perspective transform* dimana setiap citra sumber di transformasikan ke suatu bidang yang sama. Matrix 3×3 yang didapat dari proses transformasi, digunakan untuk kalibrasi kamera. Pada gambar 3.22 merupakan proses kalibrasi, dimana untuk pengujian kalibrasi, dengan menentukan titik awal pada citra sumber 1.



(a) Citra sumber

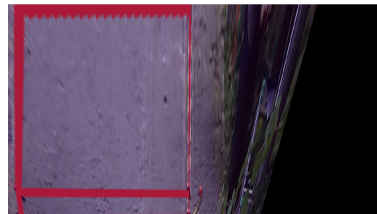


(b) Citra Hasil

Gambar 3.20: Proses *perspective transform* pada citra 1



(a) Citra sumber

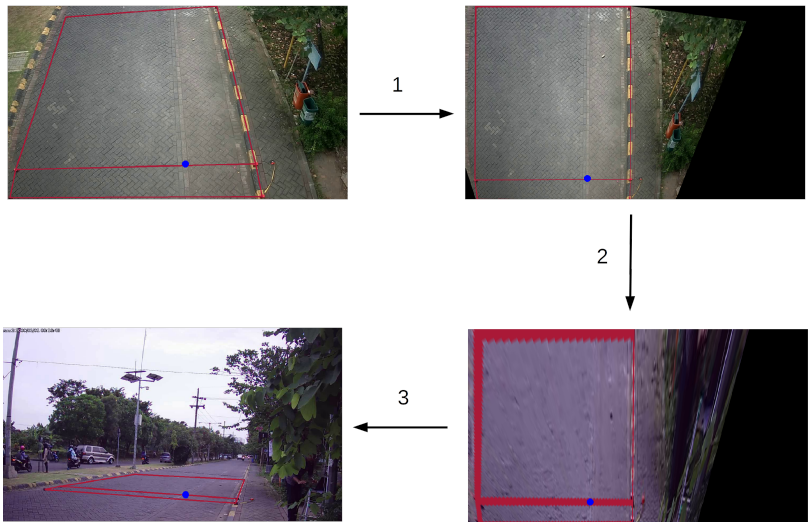


(b) Citra Hasil

Gambar 3.21: Proses *perspective transform* pada citra 2

Untuk mengilustrasikannya dengan cara menggambar sebuah lingkaran pada citra sumber 1 dengan titik koordinat (x,y) yang sudah ditentukan. Matrix 3×3 yang didapat dari transformasi citra sumber satu dikalikan dengan titik koordinat (x,y) tersebut, dimana hasil nya adalah sebuah titik koordinat baru (x_n,y_n) . Titik koordinat yang didapat (x_n,y_n) di petakan pada citra hasil transformasi dari citra sumber 1 dengan menggambar sebuah lingkaran. Posisi penanda pada kedua citra hasil transformasi dari citra sumber

1 dan citra sumber 2 terlihat sama, maka titik koordinat (x_n, y_n) yang di petakan pada citra hasil 1 juga dipetakan pada citra hasil 2. Matrix 3x3 yang didapat dari proses transformasi citra sumber 2 dilakukan perhitungan invers matrix 3x3. Hasil yang diperoleh dari proses invers dikalikan dengan titik koordinat (x_n, y_n) , dimana akan menghasilkan sebuah titik koordinat baru (x_m, y_m) . Selanjutnya titik koordinat (x_m, y_m) dipetakan pada citra sumber 2. Berdasarkan gambar 3.22 terlihat bahwa lokasi gambar lingkaran pada citra sumber 1 hampir sama dengan lokasi gambar lingkaran pada citra sumber 2.



Gambar 3.22: Diagram alir proses *perspective transform*.

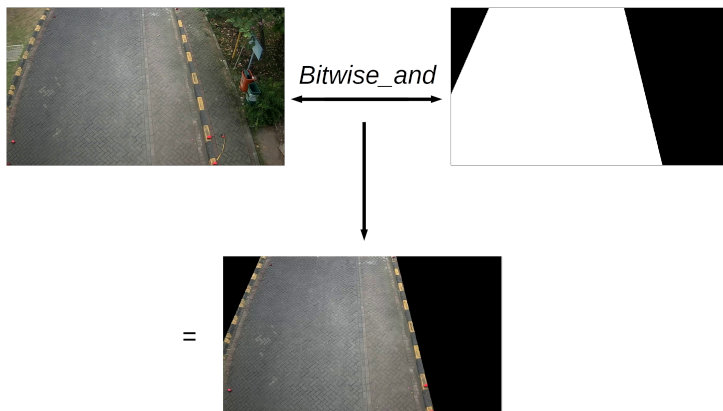
3.5 Penjejakan Kendaraan

Pada subbab ini yaitu menjelaskan proses penjejakan kendaraan. Penjejakan bertujuan untuk mendapatkan posisi koordinat tiap kendaraan yang bergerak dalam video masukan. Video masukan yang di proses pada tahap ini yaitu video masukan 1 yang ditangkap oleh kamera dari posisi atas. Penjejakan kendaraan dilakukan dengan menggunakan metode *matching-based*. Tahapan ini

merupakan tahapan yang paling penting dalam penelitian ini karena sebagian besar proses dilakukan pada tahap ini.

3.5.1 Seleksi Area

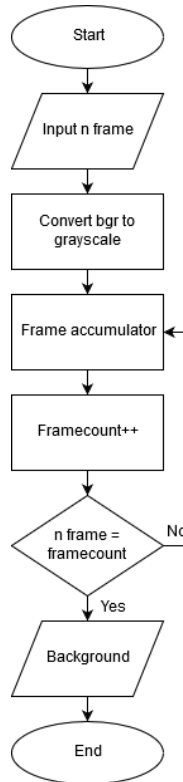
Pada tahap ini bertujuan untuk menyeleksi video masukan yaitu pada area mana saja yang akan diproses pada tahap selanjutnya. Pada tahap akuisisi data terlihat bahwa area yang ditangkap oleh kamera 1 tidak hanya area jalan saja tetapi juga area sekitarnya. Dengan menyeleksi area yang akan diproses selanjutnya, diharapkan proses pada tahap selanjutnya dapat berlangsung lebih cepat dan mengurangi gangguan yang disebabkan oleh objek yang terdeteksi pada area selain area jalan. Metode yang digunakan untuk menyeleksi area menggunakan *bitwise and operator* [20], dimana hasil yang didapatkan yaitu menampilkan area berdasarkan bentuk yang dimasukkan. Berdasarkan kondisi jalan pada citra yaitu berbentuk poligon dengan lima sisi, maka dibutlah sebuah citra yang memiliki bentuk poligon lima sisi dengan warna putih untuk tiap pikselnya dimana setiap titik dari pepotongan antar sisi sesuai dengan titik koordinat area jalan pada citra masukan 1. Pada gambar 3.23 merupakan proses seleksi area menggunakan *bitwise and operator*.



Gambar 3.23: Proses Seleksi Area Menggunakan *Bitwise And*.

3.5.2 Pembuatan citra Latar

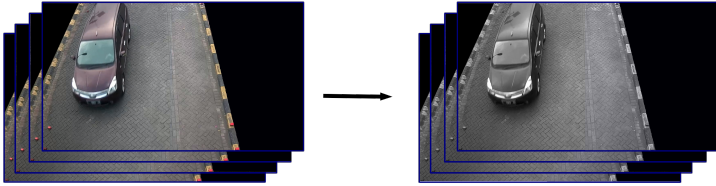
Pembuatan citra latar atau *Background Reconstruction* adalah cara untuk membentuk citra latar (*background*) dari suatu citra berjalan atau video. Jenis *background reconstruction* yang digunakan adalah *running average*. Pada gambar 3.24 merupakan *flowchart* dari proses pembuatan citra latar.



Gambar 3.24: Diagram alir proses pembuatan citra latar.

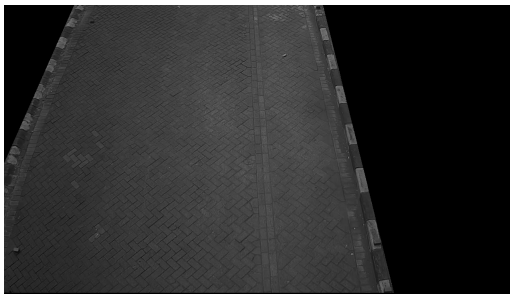
Citra hasil proses seleksi sebelumnya memiliki ruang warna *bgr*, dimana harus konversi ke ruang warna abu-abu (*grayscale*). proses konversi ruang warna *bgr* ke *grayscale* dilakukan secara terus menerus sebanyak citra masukan.

Pada gambar 3.25 merupakan proses konversi ruang warna *bgr* ke *grayscale*. Pembuatan citra latar dilakukan dengan menghitung jumlah bobot nilai piksel dari citra masukan kemudian dilakukan perhitungan rata-rata dari kumpulan citra masukan.

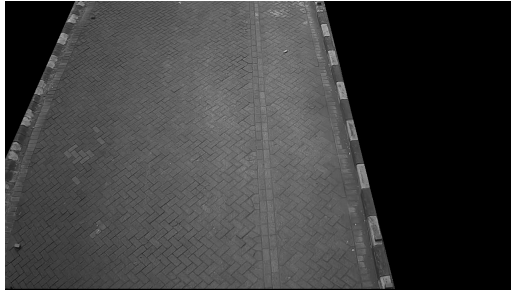


Gambar 3.25: Proses konversi ruang warna *bgr* ke *grayscale*.

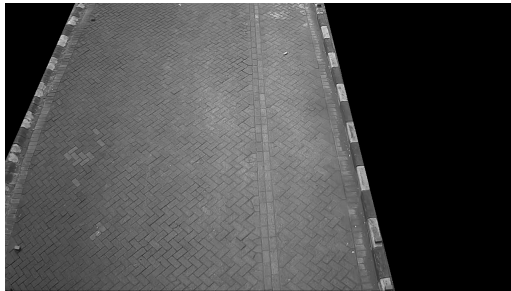
Pada gambar 3.26, gambar 3.27, gambar 3.28 merupakan hasil dari proses *background reconstruction* dengan jumlah citra masukan yang berbeda-beda. Pada gambar 3.26, citra masukan yang diproses sebanyak 75 citra sehingga menghasilkan citra latar yang cukup gelap. Jumlah citra masukan pada gambar 3.27 yaitu sebanyak 150 citra sehingga jika diperhatikan jelas terlihat lebih terang dari pada citra latar dengan masukan 75 citra. Hasil dari proses pembuatan citra latar yang dirasa cukup bagus sebagai sebuah citra latar yaitu dengan memasukkan 300 citra masukan, dimana pada gambar 3.28 terlihat bahwa citra latar yang dihasilkan cukup terang dibandingkan citra latar dengan masukan 75 atau 150 citra.



Gambar 3.26: Citra hasil proses *background reconstruction* dengan jumlah citra masukan sebanyak 75 citra.



Gambar 3.27: Citra hasil proses *background reconstruction* dengan jumlah citra masukan sebanyak 150 citra.



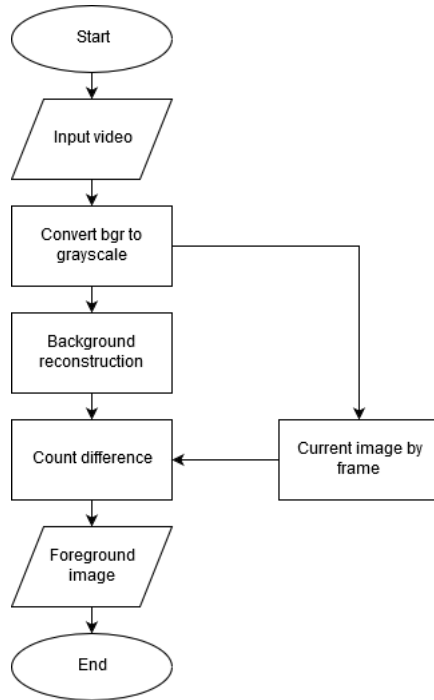
Gambar 3.28: Citra hasil proses *background reconstruction* dengan jumlah citra masukan sebanyak 300 citra.

Berdasarkan citra hasil proses pembuatan citra latar dengan masukan jumlah citra yang berbeda-beda, semakin banyak citra masukan maka hasilnya akan semakin terang dan tajam. Dalam penelitian ini jumlah citra masukan yang digunakan sebanyak 300 citra. Citra latar yang sudah didapat akan tersimpan ke penyimpanan lokal.

3.5.3 Substraksi Citra Latar

Pada tahap ini yaitu substraksi citra latar atau *background subtraction* bertujuan untuk mendapatkan citra latar depan (*foreground*). Jenis substraksi citra latar yang digunakan pada proses ini yaitu *frame difference*, dimana membutuhkan dua buah citra yai-

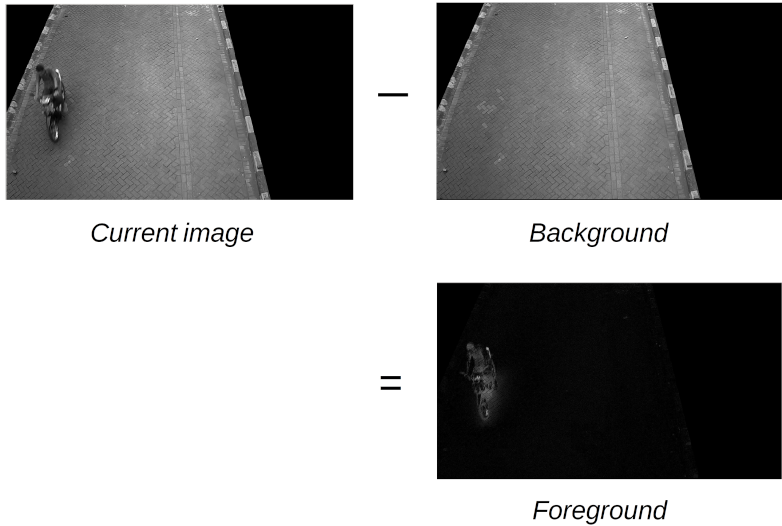
tu citra latar (*background*) dan citra sekarang untuk dibandingkan sehingga menghasilkan sebuah citra latar depan. Proses kerja dari *background subtraction* adalah dengan mengurangi antara citra masukan dengan citra latar. Fungsi yang digunakan dalam proses ini yaitu *absdiff()* yang berfungsi untuk menghitung perbedaan absolut antara citra *frame* sekarang dengan citra latar. Pada gambar 3.29 merupakan diagram alir dari proses *background subtraction*.



Gambar 3.29: Diagram alir proses pembuatan citra latar depan (*Background subtraction*).

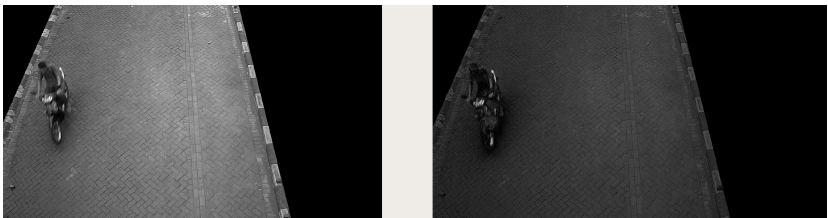
Pada proses subtraksi citra ini dilakukan dengan operasi pengurangan antara citra sekarang dan citra referensi. Pada gambar 3.30 merupakan ilustrasi dari proses subtraksi citra dengan objek kendaraan roda dua, dimana citra masukan berupa *current frame* atau citra pada waktu ($t+1$) dikurangi dengan citra latar dan meng-

hasilkan sebuah citra latar depan atau *foreground*.



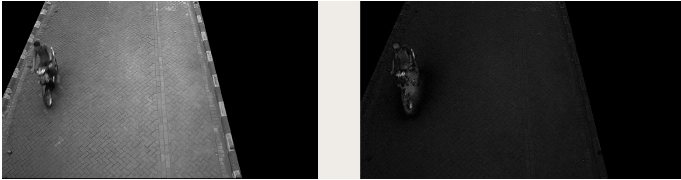
Gambar 3.30: Citra hasil *Background subtraction*.

Jumlah citra masukan saat proses pembuatan citra latar mempengaruhi hasil dari proses *background subtraction*. Berdasarkan gambar 3.31 ini dilakukan percobaan *background subtraction* dengan citra masukan 75 citra.

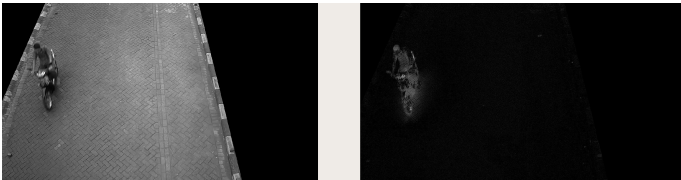


Gambar 3.31: Citra hasil proses *background subtraction* menggunakan citra latar dengan jumlah citra masukan sebanyak 75 citra.

Berdasarkan gambar 3.32 dan gambar 3.33, citra masukan yang digunakan secara berurutan yaitu 150 citra dan 300 citra. Berdasarkan Setiap percobaan yang dilakukan, hasil yang didapat menunjukkan bahwa citra latar dengan masukan jumlah paling besar menghasilkan citra *foreground* lebih jelas dari pada hasil yang didapat menggunakan citra latar dengan masukan citra lebih sedikit.



Gambar 3.32: Citra hasil proses *background subtraction* menggunakan citra latar dengan jumlah citra masukan sebanyak 150 citra.



Gambar 3.33: Citra hasil proses *background subtraction* menggunakan citra latar dengan jumlah citra masukan sebanyak 300 citra.

3.5.4 Substraksi Citra Biner

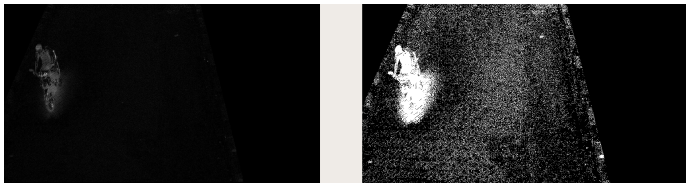
Citra *foreground* yang didapat dari proses *background subtraction* masih memiliki derajat nilai warna keabuan yang beragam. Agar citra *foreground* tersebut dapat dengan mudah diproses pada tahap selanjutnya, maka diperlukannya proses substraksi citra biner atau *thresholding* agar citra *foreground* memiliki ruang warna hitam-putih atau *binary image*.

Proses *thresholding* dilakukan dengan cara mengubah nilai piksel pada citra masukan dengan nilai piksel yang diinginkan. Nilai tujuan dari proses *thresholding* ini adalah 255 atau berwarna putih. nilai piksel pada citra masukan diubah menjadi nilai piksel yang ba-

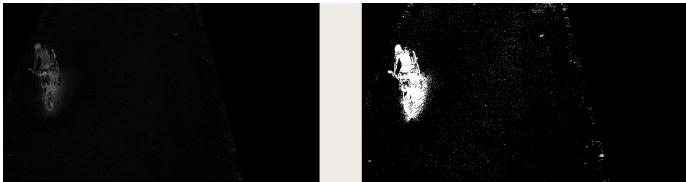
ru jika nilai piksel pada koordinat (x,y) melebihi nilai ambang atau *thresh value*.

Percobaan nilai ambang (*thresh*) dilakukan agar mengetahui pengaruh derajat nilai warna keabuan yang digunakan sebagai nilai ambang batas terhadap binerisasi citra. Data masukan yaitu berupa citra *foreground* yang merupakan hasil dari proses *background subtraction*. Dalam penelitian ini nilai ambang batas yang digunakan yaitu sebesar tiga puluh pada kondisi sore hari. Percobaan masukan nilai ambang batas yang dilakukan yaitu sepuluh, dua puluh, dan tiga puluh.

Berdasarkan gambar 3.34, hasil dari proses *thresholding* dengan nilai ambang batas sepuluh terlihat terdapat *noise* yang bermunculan. Sedangkan pada gambar 3.35, dimana nilai ambang yang digunakan sebesar dua puluh terlihat bahwa *noise* yang bermunculan lebih sedikit dari citra hasil sebelumnya tetapi *noise* yang muncul masih terlihat cukup banyak. Penetapan nilai ambang sebesar tiga puluh, menghasilkan citra yang memiliki *noise* cukup sedikit.



Gambar 3.34: Citra hasil proses *thresholding* dengan nilai ambang sepuluh.



Gambar 3.35: Citra hasil proses *thresholding* dengan nilai ambang dua puluh.

Pada gambar 3.36, nilai ambang yang digunakan yaitu tiga puluh, dimana citra yang dihasilkan terlihat bahwa sedikit *noise* yang bermunculan dari pada citra hasil *threshold* dengan nilai ambang dua puluh.



Gambar 3.36: Citra hasil proses *thresholding* dengan nilai ambang tiga puluh.

Proses *thresholding* dilakukan dengan cara mengubah nilai piksel pada citra masukan dengan nilai piksel yang diinginkan. Nilai tujuan dari proses *thresholding* ini adalah 255 atau berwarna putih. nilai piksel pada citra masukan diubah menjadi nilai piksel yang baru jika nilai piksel pada koordinat (x,y) melebihi nilai ambang atau *thresh value*.

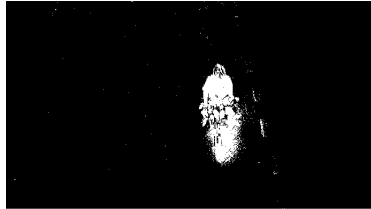
3.5.5 Operasi Morfologi

Operasi morfologi atau *morphological operation* merupakan bentuk filter yang berguna untuk membersihkan *noise* pada citra hasil proses substraksi citra biner [9]. Dalam penelitian ini digunakan dua macam *morphological operation* yaitu *erosion* dan *dilation*.

Erosi merupakan metode *morphological operation* dasar pada citra digital yang berfungsi untuk mengurangi nilai piksel dengan membandingkan nilai piksel terhadap nilai piksel tetangganya. Pada penelitian ini, proses erosi dilakukan dengan jumlah iterasi yang digunakan sebanyak tiga.

Dilasi merupakan metode *morphological operation* dasar pada citra digital yang berfungsi untuk menambah nilai piksel dengan membandingkan nilai piksel terhadap nilai piksel tetangganya. Proses dilasi sering digunakan untuk menutup piksel yang "bolong" pada sebuah citra digital. Pada penelitian ini iterasi yang digunakan pada proses dilasi sebanyak sembilan.

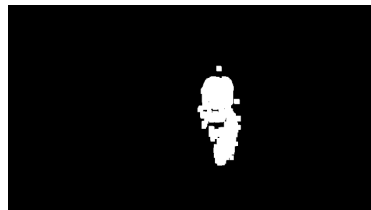
Pada gambar 3.37 ini merupakan citra hasil *thresholding* dimana akan di jadikan sebagai citra masukan dalam proses erosi atau *erosion*. Pada penelitian ini, proses erosi yang dilakukan menggunakan ukuran *kernel matrix* (3x3) dengan jumlah iterasi sebanyak tiga. Hasil dari proses erosi atau *erosion* dapat dilihat pada gambar 3.8 dan hasil dari proses dilasi terdapat pada gambar 3.9.



Gambar 3.37: Citra hasil proses *thresholding*.

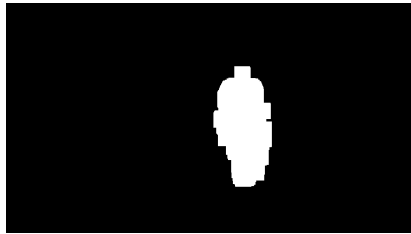


Gambar 3.38: Citra hasil proses *erosion* dengan nilai iterasi sejumlah tiga.



Gambar 3.39: Citra hasil proses *dilation* dengan nilai iterasi sejumlah lima.

Kernel yang digunakan pada citra hasil proses dilasi yaitu dengan *kernel* (3,3), dimana citra masukan yang di proses adalah citra hasil proses erosi. Pada citra tersebut terlihat "lubang" yang terdapat pada citra hasil erosi menjadi berkurang, tetapi masih ada kumpulan piksel dengan warna putih yang masih terpisah-pisah. Agar mendapatkan hasil proses segmentasi yang optimal, maka dilakukannya proses erosi dan dilasi secara berkali-kali hingga mendapatkan hasil yang diinginkan. Dengan lima kali perulangan, bentuk yang dihasilkan terlihat cukup baik yang seperti pada gambar 3.40.



Gambar 3.40: Citra hasil proses *erosion* dan *dilation* sebanyak lima kali.

3.5.6 Deteksi Kendaraan

Deteksi kendaraan merupakan tahapan untuk mencari area-area yang diduga sebagai kendaraan bermotor. Dalam mencari area yang diduga sebagai kendaraan, cara yang dapat digunakan adalah mencari kontur atau *contour*. Kontur merupakan keadaan yang ditimbulkan oleh perubahan intensitas pada piksel-piksel yang bertetangga. Adanya perubahan intensitas tersebut, tepi pada citra *foreground* dapat dideteksi. Dengan terdeteksinya tepi tepi tiap objek, maka fitur tersebut dalam digunakan untuk mencari posisi tiap objek yang terdeteksi. Untuk menemukan kontur dari sebuah objek, dapat menggunakan fungsi *findContours()*. Setelah menemukan kontur dari tiap objek, agar dapat di ilustrasikan, maka digambarkan kontur tersebut menggunakan fungsi *drawContours()*. Hasil dari deteksi kontur dan menggambar kontur tersebut dapat dilihat pada gambar 3.41.

Berdasarkan gambar 3.41, area yang sudah terpasang dengan kontur, sudah dapat diduga sebagai kendaraan bermotor. Area-



Gambar 3.41: Citra hasil proses operasi morfologi dan Deteksi kontur.

area yang telah diperoleh sebagai area yang dianggap kendaraan bermotor dapat dicari posisi tiap-tiap area tersebut dalam koordinat (x, y) . Agar dapat menemukan posisi pada tiap area yang terkontur dapat menggunakan momen atau *moment*.

Moment dapat menggambarkan suatu objek dalam hal area, posisi, orientasi dan parameter terdefinisi lainnya. Persamaan dasar dari momen suatu objek didefinisikan pada persamaan 3.1.

$$m_{ij} = \sum_x \sum_y x^i y^j a_{xy} \quad (3.1)$$

Dengan,

$$\begin{aligned} i, j &= \text{Orde dari momen} \\ x, y &= \text{Koordinat titik} \\ a_{xy} &= \text{Intensitas titik} \end{aligned}$$

Momen tingkat ke-0 dan ke-1 (*zero-and first-order moments*) didefinisikan pada persamaan 3.2, 3.3.

$$m_{00} = \sum_x \sum_y a_{xy} \quad (3.2)$$

$$m_{10} = \sum_x \sum_y x a_{xy} \quad (3.3)$$

Momen tingkat ke-0 dan ke-1 (*zero-and first-order moments*) didefinisikan pada persamaan 3.4.

$$m_{01} = \sum_x \sum_y y a_{xy} \quad (3.4)$$

Pada citra biner yang mana a_{xy} bernilai 0 atau 1, momen tingkat ke-0 (m_{00}) sama dengan area dari objek. Pusat area dari setiap kendaraan yang terdeteksi ditentukan dalam (x,y) , dimana pusat area dari setiap objek didefinisikan pada persamaan 3.5.

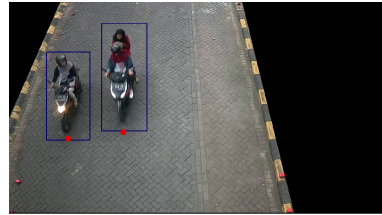
$$x^p = \frac{m_{10}}{m_{00}} \text{ dan } y^p = \frac{m_{01}}{m_{00}} \quad (3.5)$$

(x^p, y^p) merupakan pusat koordinat dari objek yang terdeteksi [21]. Posisi koordinat x dan y yang diperoleh dijadikan titik tengah untuk menggambar sebuah lingkaran sebagai ilustrasi posisi tiap objek yang terdeteksi seperti yang terlihat pada gambar 3.42 (a).

Pada penelitian ini, cangkupan citra yang sanggup ditangkap kamera 1, tidaklah cukup luas, sehingga penentuan titik pusat area sebagai acuan untuk menentukan lokasi kuranglah cukup. Jadi acuan titik yang mengilustrasikan lokasi tiap objek terdeteksi di letakan pada ujung kendaraan seperti pada gambar 3.42 (b). Untuk mengetahui panjang dan lebar dari objek yang terdeteksi maka menambahkan *bounding box* pada setiap objek yang terdeteksi. Pada gambar 3.43 merupakan diagram alir dalam proses menggambar *circle* dan *retacgle* pada objek yang terdeteksi.

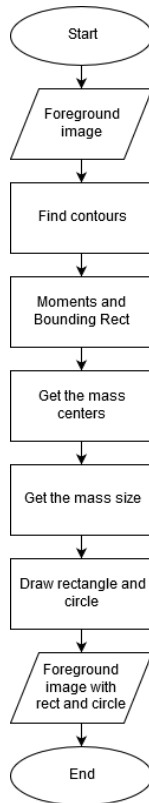


(a) Pada titik pusat



(b) Pada ujung kendaraan

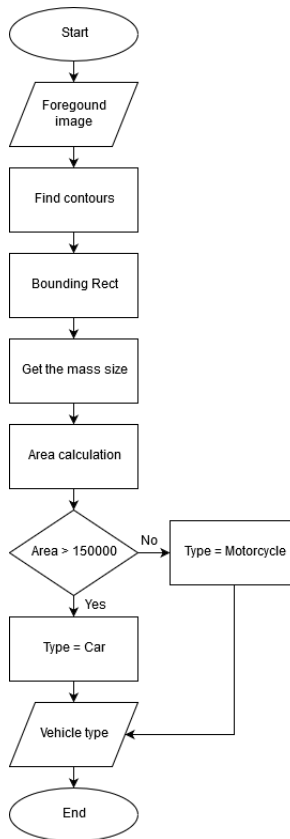
Gambar 3.42: Menggambar *circle* pada objek yang terdeteksi



Gambar 3.43: Diagram alir menggambar *circle* dan *rectangle* pada objek yang terdeteksi.

3.5.7 Deteksi Jenis Kendaraan

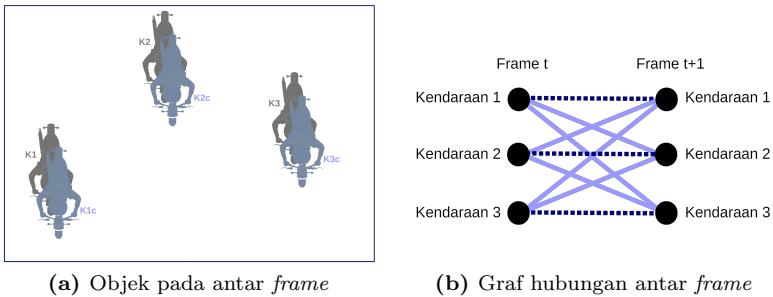
Proses deteksi jenis kendaraan dilakukan dengan membandingkan luasan objek yang terdeteksi [15] dengan nilai ambang. Luasan tiap objek didapatkan dengan mengkalikan ukuran panjang dan lebar dari *bounding box* yang sudah didapatkan pada proses deteksi kendaraan. Pada gambar 3.44 merupakan diagram alir proses deteksi jenis kendaraan.



Gambar 3.44: Diagram alir deteksi jenis kendaraan pada objek yang terdeteksi.

3.5.8 Proses Penjejakan Kendaraan

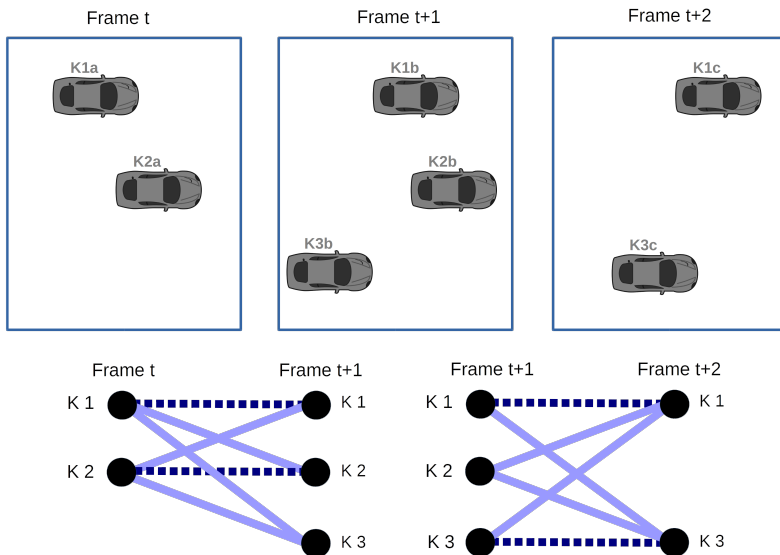
Proses penjejakan kendaraan menggunakan algoritma *matching-based*. *Matching-based* merupakan algoritma untuk menemukan kesamaan terbesar dari beberapa titik berdasarkan nilai kesamaan terbesar dari titik yang dicari. Titik disini merupakan titik koordinat yang sudah didapat pada proses sebelumnya untuk mengilustrasikan posisi kendaraan dimana titik acuan nya diletakan pada ujung setiap objek yang terdeteksi. Pada penelitian ini fitur kesamaan yang digunakan yaitu nilai jarak dari titik pada *frame* sekarang ke *frame* berikutnya. Perhitungan jarak tiap titik antar *frame* menggunakan persamaan *euclidean*. Berdasarkan gambar 3.45, penggunaan algoritma *matching-based* pada penelitian ini digunakan untuk menemukan fitur kesamaan terbesar atau nilai *euclidean* terkecil dari titik koordinat antara *frame* sekarang dan *frame* selanjutnya [14].



Gambar 3.45: Pencarian hubungan antar *frame*

Setiap objek yang diduga sebagai kendaraan bermotor memiliki fitur lokasi dalam (x,y) masing-masing, dimana proses untuk mendapatkan titik koordinat terdapat pada proses deteksi kendaraan. Pada proses penjejakan kendaraan ini, jika terdapat objek atau kendaraan yang terdeteksi, maka titik koordinat dari objek tersebut akan di simpan dalam sebuah variabel *array* sementara pada *frame t*, dimana tujuannya yaitu sebagai data pembanding untuk titik koordinat pada *frame t+1*. Jika titik koordinat pada *frame t* sudah tersimpan pada variabel *array* sementara, maka pada *frame t+1* akan dilakukan perhitungan *euclidean*, dimana pada proses perhitungan tersebut yaitu menghitung nilai jarak antara dua titik

koordinat (x,y) . Dua titik yang dimaksud yaitu titik objek pada *frame t+1* dengan titik objek pada *frame t*. Jika hasil dari perhitungan *euclidean* tersebut menghasilkan nilai di bawah nilai ambang yang sudah ditetapkan, maka objek pada *frame t+1* merupakan objek yang sama dengan objek pada *frame t*. Berdasarkan percobaan yang dilakukan berulang-ulang, nilai ambang batas yang ditetapkan yaitu sebesar lima puluh lima. Setelah menetapkan bahwa objek tersebut pada *frame t+1* adalah satu objek yang sama dengan objek pada *frame t*, maka nilai titik pada data *array* akan *update* dengan nilai titik pada *frame t+1*. Proses tersebut akan terus berulang hingga objek tersebut tidak terdeteksi dan jika objek tersebut sudah tidak terdeteksi maka nilai titik koordinat objek tersebut yang terdapat pada variabel *array* akan dihapus. Berdasarkan gambar 3.46 terlihat bahwa terdapat lebih dari satu objek yang terdeteksi.

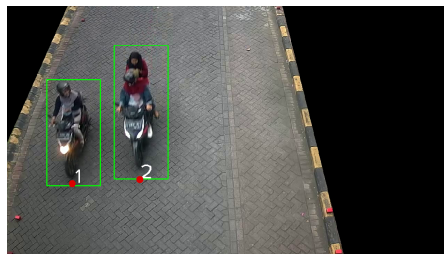


Gambar 3.46: Graf hubungan antar frame.

Jika pada saat *frame t* terdapat satu objek yang terdeteksi dan pada saat *frame t+1* terdapat dua objek yang terdeteksi maka pada saat *frame t+1* setiap koordinat akan dilakukan perhitungan

euclidean dengan titik koordinat pada *frame t*. Karena pada *frame t+1* terdapat dua titik koordinat maka dilakukan pengecekan secara berkala. Jika titik koordinat pertama pada *frame t+1*, hasil perhitungan *euclidean* berada dibawah nilai ambang batas, maka objek dengan titik koordinat tersebut adalah objek pertama atau sama dengan objek pada *frame t*, dan jika pada titik koordinat kedua pada *frame t+1*, hasil perhitungan *euclidean* berada di atas nilai ambang batas. maka objek dengan titik koordinat tersebut merupakan sebuah objek baru dan titik koordinat objek tersebut akan di simpan pada variabel *array* yang baru. Pada frame selanjutnya perhitungan *euclidean* dilakukan dengan dua data pembanding selama jika yang terdeteksi hanya kedua objek tersebut saja. Proses tersebut akan berlaku jika terdapat objek baru yang terdeteksi lagi saat kedua objek sebelumnya masih terdeteksi.

Pada gambar 3.46 juga terlihat adanya objek atau kendaraan yang sudah tidak terdeteksi pada *frame t+2*. Sesuai dengan kondisi tersebut jika pada *frame t+1* masih terdeteksi adanya tiga kendaraan dan pada *frame t+2* hanya terdeteksi dua kendaraan, maka pada *frame t+2* akan tetap melakukan perhitungan *euclidean* terhadap tiga data pembanding yang terdapat pada *frame t+1*. Jadi sesuai yang diilustrasikan identitas tiap kendaraan tidak berubah meskipun ada objek lain yang sudah tidak terdeteksi. jika tidak ada objek yang terdeteksi maka semua data titik koordinat yang berada pada variabel *array* yang digunakan sebagai data pembanding untuk frame selanjutnya akan dihapus semua. Pada gambar 3.47 merupakan citra hasil proses penjejukan.



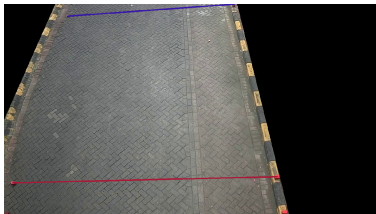
Gambar 3.47: Hasil penjejukan kendaraan bermotor.

Pada gambar 3.49 merupakan diagram alir proses penjejakan. Berdasarkan citra hasil proses penjejakan, dimana pada citra tersebut terlihat ada penomoran untuk tiap objek yang terdeteksi.

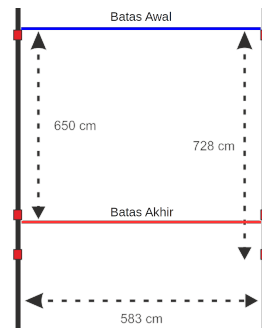
3.6 Perhitungan Kecepatan Kendaraan

Pada subab ini yaitu menjelaskan proses perhitungan kecepatan kendaraan yang melintas. Pada tahap kalibrasi kedua kamera, dimana citra hasil tangkapan kamera 1 telah ditransformasikan ke bidang lain sesuai kondisi nyatanya dalam satuan *centimeter*. Hasil dari proses kalibrasi tersebut adalah mendapatkan sebuah matrix 3x3. Matrix yang didapat dari proses transformasi pada citra 1, akan digunakan dalam proses perhitungan kecepatan pada tahap ini. Dengan mengkalikan setiap titik koordinat yang dimiliki oleh objek yang terdeteksi dengan matrix 3x3 tersebut, dapat diasumsikan bahwa titik koordinat posisi kendaraan yang terdeteksi pada citra seolah-olah berada pada kondisi nyata dengan titik koordinat baru hasil dari perkalian matrix sebelumnya.

Langkah selanjutnya yaitu menentukan jarak yang harus ditempuh setiap kendaraan yang terdeteksi dan juga menentukan letak batas awal dan batas akhir yang harus dilalui setiap kendaraan yang terdeteksi seperti pada gambar 3.48.

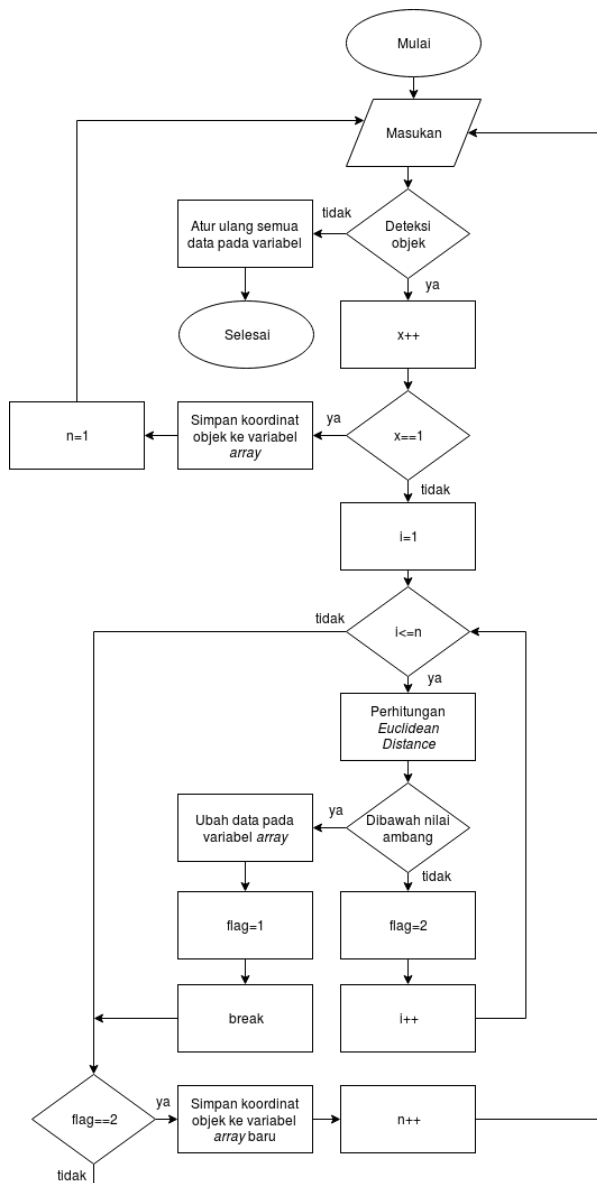


(a) Batas jarak pada citra



(b) Batas jarak pada ilustrasi

Gambar 3.48: Batas awal dan batas akhir dalam penentuan jarak tempuh



Gambar 3.49: Diagram alir proses penjejakan.

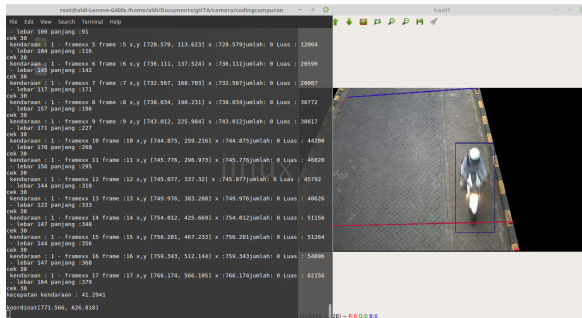
Berdasarkan hasil pengukuran, terlihat bahwa jarak antara batas awal dan batas akhir yang harus dilalui sepanjang 650 *centimeter*. Berdasarkan persamaan untuk perhitungan kecepatan kendaraan, dimana membutuhkan beberapa data yaitu jarak tempuh, kecepatan tangkap kamera atau fps (*frame per second*), dan jumlah *frame* yang ditempuh. Jumlah *frame rate* yang digunakan yaitu sebesar 30 fps. Selama kendaraan yang terdeteksi berada pada batas awal dan batas akhir, pada saat itulah di hitung ada berapa banyak *frame* yang sudah ditempuh. Proses perhitungan kecepatan setiap kendaraan yang terdeteksi, di proses saat kendaraan tersebut telah melebihi batas akhir. Hasil yang didapat dari proses perhitungan tersebut langsung dikonversi ke satuan *kilometer per jam*.

3.7 Pelanggaran Kecepatan

Nilai kecepatan yang sudah didapat dalam proses sebelumnya dapat di jadikan sebagai data input untuk ditentukan apakah terjadi pelanggaran kecepatan atau tidak. dengan menetapkan nilai batas kecepatan yang harus dipatuhi sesuai peraturan setempat yang ada, maka setiap data kecepatan kendaraan yang melintas dapat di seleksi. Pada penelitian ini berada di lingkungan kampus ITS, dimana dalam pengujian nilai batas maksimal kecepatan yang harus dipatuhi yaitu sebesar 40 *kilometer per hours*. Pada Gambar 3.50 terlihat adanya pengendaraan yang melanggar kecepatan berkendara, dimana nilai kecepatan yang didapat yaitu 41 Km/jam.

3.7.1 Tangkap Citra Pertama Pelanggar

Jika terjadi pelanggaran kecepatan pada kendaraan yang terdeteksi, maka sistem akan langsung mengambil citra kendaraan tersebut. Setiap objek atau kendaraan yang terdeteksi telah digambar sebuah *rectangle* sesuai dengan ukuran tiap kendaraan yang terdeteksi. Setiap kendaraan yang terdeteksi juga telah dilakukan proses penjejakan dimana setiap objek yang terdeteksi memiliki identitas masing masing, jadi misalkan ada lebih dari satu kendaraan yang melintas jika terdapat satu kendaraan yang melakukan pelanggaran, maka citra dari kendaraan yang melanggar tersebut akan di tangkap dan disimpan dalam sebuah file lokal.



Gambar 3.50: Percobaan Pelanggaran Kecepatan.

Berdasarkan gambar 3.51, terlihat ada dua citra dimana kedua citra tersebut melakukan pelanggaran kecepatan, sehingga sistem akan mengambil citra dari setiap pelanggar kecepatan dan disimpan pada penyimpanan lokal. Saat proses pengambilan citra pelanggar, sistem ini juga akan mengambil titik koordinat kendaraan tersebut yang akan digunakan sebagai data masukan pada proses selanjutnya yaitu tangkap citra kedua pelanggar.



(a) Citra mobil



(b) Citra sepeda motor

Gambar 3.51: Tangkap citra pelanggar kecepatan

3.7.2 Tangkap Citra Kedua Pelanggar

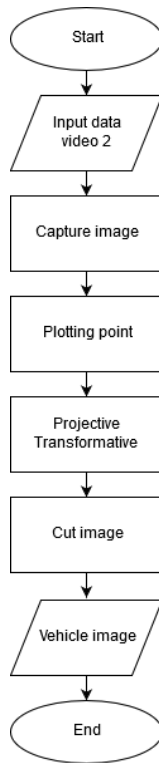
Pada proses ini yaitu menangkap gambar kendaraan yang melaju melebihi kecepatan yang telah ditentukan dari posisi depan. Terdapat beberapa proses yang dilakukan untuk mendapatkan citra dari kendaraan yang melanggar. Pada proses ini telah dilakukan pemasangan penanda dan sebuah pipa di sisi kanan dan kiri jalan. pada gambar 3.52 merupakan ilustrasi hasil pemasangan penanda pipa.



Gambar 3.52: Ilustrasi pemasangan penanda pipa

Pemasangan pipa dan pananda pada citra hasil tangkapan kamera 2 bertujuan agar mengetahui orientasi kemiringan yang dihasilkan citra tersebut, sehingga dapat diproses agar mendapatkan citra yang lebih lurus. Setiap titik koordinat dari penanda dan pipa yang digunakan, disimpan pada suatu variabel. Penentuan titik koordinat dari penanda dan pipa yang digunakan, dilakukan secara manual. Titik koordinat (x,y) yang didapat akan digunakan sebagai data masukan proses transformasi citra pada proses selanjutnya.

Berdasarkan diagram alir pada gambar 3.53. Data masukan yang digunakan yaitu citra tangkap kamera 2, dimana saat kamera 2 melakukan proses penangkapan video, sistem akan langsung mengekstrak citra masukan menjadi satu citra dan disimpan pada penyimpanan lokal. Pada gambar 3.54 merupakan citra hasil ekstraksi pada citra masukan kamera 2.



Gambar 3.53: Diagram alir proses penjejakan.



Gambar 3.54: Citra pelanggar hasil tangkapan kamera kedua.

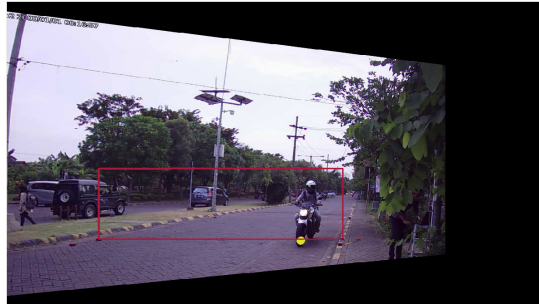
Berdasarkan proses kalibrasi kedua kamera pada tahap sebelumnya, dimana sudah berhasil dilakukan proses kalibrasi dan melakukan percobaan *plotting* dari sebuah titik koordinat masukan. Pada proses tangkap citra kedua pelanggar ini, dimana adanya proses *plotting*, titik koordinat masukan berasal dari titik koordinat yang didapat pada proses sebelumnya yaitu tangkap citra pertama pelanggar. Pada gambar 3.55 merupakan hasil proses *plotting*.



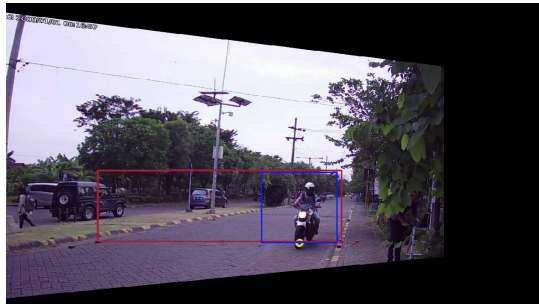
Gambar 3.55: Citra hasil proses *plotting*.

Setelah proses *plotting* dilakukan, langkah selanjutnya dilakukan proses transformasi citra. citra yang ditangkap kamera 2 terlihat miring, sesuai dengan data titik koordinat tiap penanda dan pipa, maka bidang tersebut ditransformasikan ke kondisi nyata dengan ukuran dalam satuan *centimeter*.

Pada gambar 3.56 terlihat bahwa ukuran penanda dan pipa pada citra terlihat lebih lurus. Pada tahap terakhir yaitu memotong citra tersebut menjadi lebih kecil sehingga yang terlihat hanya citra pengendaranya saja. Proses transformasi citra yang sudah dilakukan menghasilkan sebuah matrix 3×3 , dimana matrix tersebut akan dijadikan sebagai pengkali untuk menentukan posisi dan cakupan citra yang harus dipotong. Data masukan yang digunakan untuk dikalikan dengan matrix 3×3 adalah titik koordinat hasil proses *plotting* pada proses sebelumnya. Pada gambar 3.57 merupakan hasil dari proses perkalian matrix dimana menghasilkan sebuah titik koordinat baru untuk mem *plotting* dan menentukan area yang di potong. Pada gambar 3.58 merupakan citra hasil proses pemotongan citra pelanggar.



Gambar 3.56: Citra hasil proses transformasi citra.



Gambar 3.57: Citra hasil proses *plotting* dan penentuan area.



(a) Citra mobil

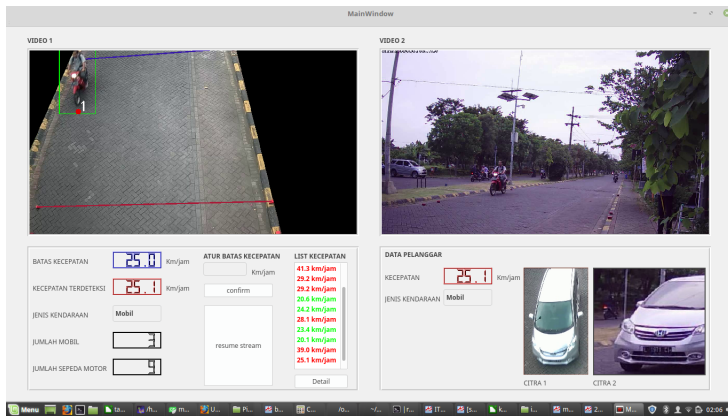


(b) Citra sepeda motor

Gambar 3.58: Tangkap citra pelanggar kecepatan

3.8 Antarmuka Pengguna Berbasis Desktop


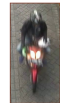


Pada dasarnya sisi *user interface* atau antarmuka pengguna dibangun secara satu kesatuan dengan pemrograman pemantauan pelanggaran kecepatan kendaraan. Setiap perhitungan kecepatan pada objek yang terdeteksi dan setiap data citra pelanggar yang tertangkap akan ditampilkan pada sisi *user interface* sekaligus data pelanggar kecepatan juga disimpan pada penyimpanan lokal dan juga pada sebuah *MySQL Database* di *local server*. *Web-server software* yang digunakan untuk menjalankan aplikasi web yaitu *Apache Webserver*. Setiap data pelanggar yang tersimpan pada *local storage* akan ditampilkan pada sebuah aplikasi web. Pada gambar 3.59 terlihat bahwa terdapat beberapa data yang ditampilkan pada sisi *user interface* yaitu video hasil tangkapan setiap *IP Camera*, data citra pelanggar kecepatan, nilai kecepatan pada setiap objek yang terdeteksi, nilai batas kecepatan, dan daftar kecepatan yang sudah dideteksi.



Gambar 3.59: Tampilan utama *user interface* pemantauan pelanggaran kecepatan.

Pada sisi *user interface* juga dapat mengubah nilai batas kecepatan, selain itu juga terdapat tombol *pause or play* dan terdapat tombol detail yang berfungsi untuk melihat daftar pelanggar kecepatan secara jelas pada aplikasi web. Data angka yang ditampilkan

pad sisi *user interface* memiliki warna yang berbeda beda yaitu pada warna merah menunjukan bahwa nilai kecepatan yang terdeteksi melebihi batas kecepatan, dimana pada data warna biru merupakan nilai batas kecepatan yang sudah ditentukan. Warna hijau menunjukan bahwa nilai kecepatan yang terdeteksi berada dibawah nilai batas kecepatan. Pada gambar 3.60 merupakan tampilan utama pada aplikasi web dimana data yang ditampilkan yaitu data pelanggan kecepatan. Konten dari data pelanggan yang ditampilkan yaitu waktu kejadian, jenis kendaraan, nilai kecepatan yang terdeteksi, dan citra pelanggan yang tertangkap.

No	Waktu	Jenis Kendaraan	Kecepatan	Foto 1	Foto 2
1	2018-05-16 00:58:31	Sepeda Motor	39 km/jam		
2	2018-05-16 00:58:32	Mobil	46.8 km/jam		

Gambar 3.60: Tampilan utama pada aplikasi web.

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi. Data yang digunakan dalam pengujian program diperoleh dengan merekam trafik lalu-lintas pada area lingkungan kampus ITS dari sudut pandang atas sejajar dan dari sudut pandang depan bawah, dimana masing masing hasil tangkapan video berdasarkan arus lalu-lintas yang mendatangi kamera. Kamera yang digunakan adalah *ip camera* dengan jumlah dua, dimana masing kamera diletakan pada posisi atas dan posisi bawah. Pengujian yang dilakukan di bagi menjadi beberapa bagian yaitu sebagai berikut :

1. Pengujian dengan Perbedaan Jumlah *frame rate*.
2. Pengujian dengan Perbedaan kategori bayangan.
3. Pengujian Jenis Kendaraan.
4. Pengujian dengan Perbedaan pemilihan ROI (Region of Interest).
5. Pengujian pelanggaran kecepatan.
6. Durasi Pengiriman Data Pelanggaran Kecepatan

4.1 Pengujian dengan Perbedaan Jumlah *Frame Rate*

Pengujian dengan perbedaan jumlah *frame rate* bertujuan untuk mengetahui pengaruh jumlah *frame rate* pada hasil perhitungan kecepatan kendaraan. Semakin besar jumlah *frame rate* yang digunakan, maka hasil video rekaman yang didapat semakin mendekati kondisi pergerakan asli. Terdapat tiga kondisi *frame rate* yang diujikan yaitu 20 *fps*, 25 *fps*, dan 30 *fps*. Resolusi yang digunakan dalam pengujian ini yaitu 1280x720 piksel. Pada pengujian ini dilakukan beberapa percobaan kecepatan pada setiap *frame rate* dan setiap kecepatan dilakukan secara berulang-ulang sebanyak tiga kali. Kecepatan yang diujikan yaitu 20 km/jam, 30 km/jam, 40 km/jam, dan 50 km/jam. Waktu pengujian dilakukan pada waktu yang sama yaitu pada saat tidak ada bayangan atau pada sore hari.

Berdasarkan hasil pengujian pada tabel 4.1, pengujian kecepatan dilakukan pada kondisi tidak ada bayangan, dimana jumlah *frame rate* yang digunakan yaitu sebesar 30 *fps*. Nilai kecepatan yang terdeteksi sesuai pada tabel 4.1 menunjukkan bahwa perbedaan kecepatan yang diuji mempengaruhi hasil dari pengujian kecepatan kendaraan oleh sistem. Pada tabel tersebut terlihat bahwa pengujian dengan nilai kecepatan 20 km/jam memiliki tingkat galat terendah yaitu sebesar 2.50%. Deteksi kecepatan dengan nilai kecepatan 40 km/jam dan 50 km/jam memiliki tingkat galat cukup tinggi. Pada pengujian kecepatan 40 km/jam memiliki galat yang paling tinggi dari pengujian kecepatan lainnya yaitu sebesar 9.86%, dimana nilai galat tersebut adalah rata rata dari nilai galat tiap kecepatan yang diuji pada kecepatan 40 km/jam.

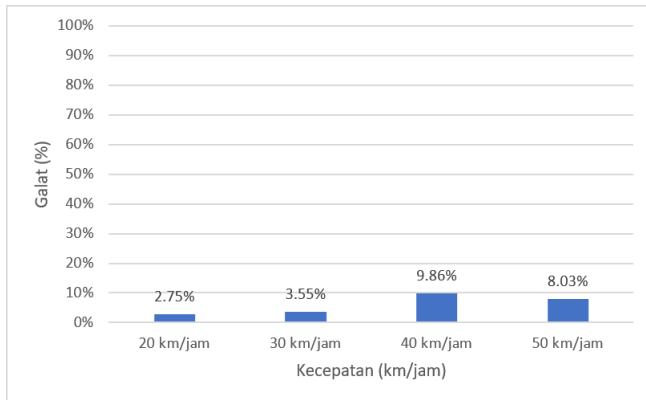
Tabel 4.1: Hasil pengujian perhitungan kecepatan kendaraan dengan *frame rate* 30 *fps*

kecepatan	Terdeteksi km/jam	Galat	frame tempuh	fps
20 km/jam	19.5	2.50%	36	30
	19.5	2.50%	36	
	20.6471	3.24%	34	
30 km/jam	30.5217	1.74%	23	
	28.08	6.40%	25	
	29.25	2.50%	24	
40 km/jam	36.9474	7.63%	19	
	35.1	12.25%	20	
	43.875	9.69%	16	
50 km/jam	46.8	6.40%	15	
	41.2941	17.41%	17	
	50.1428	0.29%	14	

Nilai galat diperoleh dari rata rata tiap nilai galat dari setiap pengujian kecepatan. Perhitungan nilai rata-rata galat untuk setiap pengujian kecepatan dapat diketahui dengan menggunakan persamaan 4.1.

$$\bar{G} = \frac{\sum_{z=1}^n \left(\frac{\text{nilaisebenarnya} - \text{nilaiterdeteksi}}{\text{nilaisebenarnya}} \right) * 100\%}{n} \quad (4.1)$$

Berdasarkan persamaan 4.1, nilai sebenarnya merupakan nilai yang diujikan yaitu kecepatan sebenarnya dan nilai terdeteksi merupakan nilai kecepatan yang terdeteksi oleh sistem. Pada gambar 4.1 merupakan grafik rata-rata galat dari setiap pengujian kecepatan dengan *frame rate* 30 *fps*.

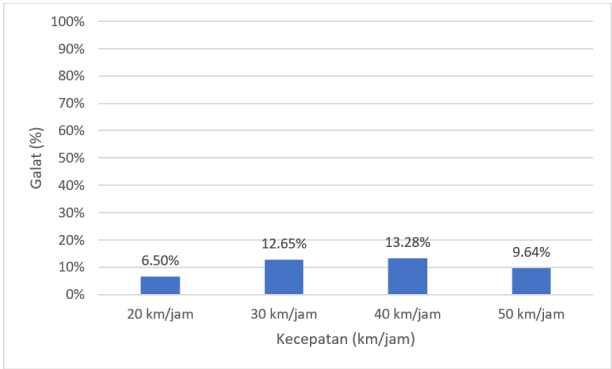


Gambar 4.1: Hasil pengujian perhitungan kecepatan dengan 30 *fps*.

Berdasarkan hasil pengujian pada tabel 4.2, pengujian kecepatan yang dilakukan dengan *frame rate* sebesar 25 *fps*. Pada pengujian ini terlihat bahwa tingkat galat yang dihasilkan lebih besar dari pada pengujian kecepatan dengan jumlah *frame rate* 30 *fps*. Pada pengujian ini juga terlihat bahwa jumlah *frame* tempuh yang dihasilkan lebih sedikit dari pada pengujian sebelumnya. Berdasarkan pada gambar 4.2, nilai galat terendah yang didapat, terdapat pada pengujian kecepatan 20 km/jam yaitu sebesar 6.50% dan galat tertinggi terdapat pada pengujian kecepatan 40 km/jam yaitu sebesar 13.28%. Selisih perbedaan nilai galat yang dihasilkan antara pengujian kecepatan dengan *frame rate* 30 *fps* dan *frame rate* 25 *fps* cukup besar. Hal ini menunjukkan tingkat akurasi yang paling tinggi dari dua percobaan yaitu pada penggunaan *frame rate* 30 *fps*.

Tabel 4.2: Hasil pengujian perhitungan kecepatan kendaraan dengan *frame rate* 25 *fps*

kecepatan	Terdeteksi km/jam	galat	frame tempuh	fps
20 km/jam	19.5	2.50%	30	25
	18.871	5.65%	31	
	17.7273	11.36%	33	
30 km/jam	26.5909	11.36%	22	
	25.4348	15.22%	23	
	26.5909	11.36%	22	
40 km/jam	36.5625	8.59%	16	
	32.5	18.75%	18	
	45	12.50%	13	
50 km/jam	48.75	2.50%	12	
	41.7857	16.43%	14	
	45	10.00%	13	



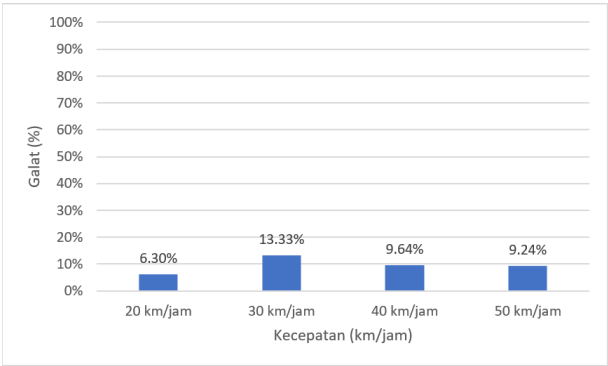
Gambar 4.2: Percobaan Pelanggaran Kecepatan dengan 25 *fps*.

Berdasarkan hasil pengujian pada tabel 4.3, pengujian kecepatan yang dilakukan dengan *frame rate* sebesar 20 *fps*. Pada pengujian ini terlihat bahwa tingkat galat yang dihasilkan lebih besar dari pada pengujian kecepatan dengan jumlah *frame rate* 30 *fps*. Pada pengujian ini juga terlihat bahwa jumlah frame tempuh yang dihasilkan lebih banyak dari pada pengujian pertama dan kedua. Berdasarkan pada gambar 4.3, nilai galat terendah yang didapat,

terdapat pada pengujian kecepatan 20 km/jam yaitu sebesar 6.30% dan galat tertinggi terdapat pada pengujian kecepatan 30 km/jam yaitu sebesar 13.33%. Selisih perbedaan nilai galat yang dihasilkan antara pengujian kecepatan dengan *frame rate* 30 *fps* dan *frame rate* 20 *fps* cukup besar. Hal ini menunjukkan tingkat akurasi yang paling tinggi dari dua percobaan yaitu pada penggunaan *frame rate* 30 *fps*.

Tabel 4.3: Hasil pengujian perhitungan kecepatan kendaraan dengan *frame rate* 20 *fps*

kecepatan	Terdeteksi km/jam	galat	frame tempuh	fps
20 km/jam	19.5	2.50%	24	20
	18.72	6.40%	25	
	18	10.00%	26	
30 km/jam	26	13.33%	18	
	26	13.33%	18	
	26	13.33%	18	
40 km/jam	39	2.50%	12	
	33.4286	16.43%	14	
	36	10.00%	13	
50 km/jam	42.5455	14.91%	11	
	46.8	6.40%	10	
	46.8	6.40%	10	



Gambar 4.3: Percobaan Pelanggaran Kecepatan.

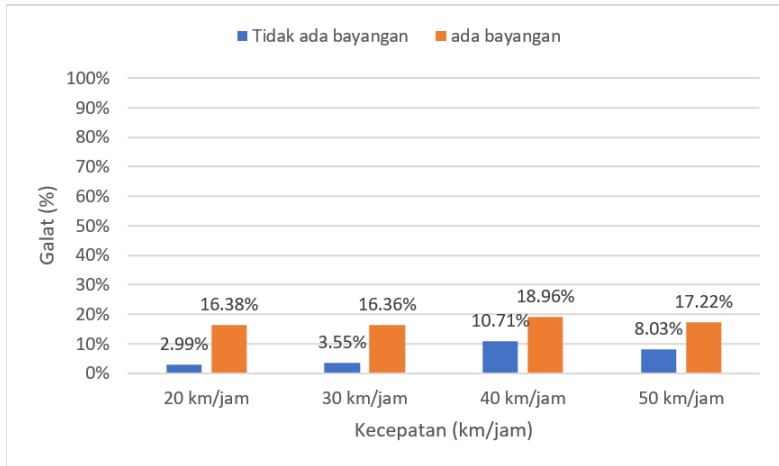
4.2 Pengujian dengan Perbedaan Kategori Bayangan

Pada tabel 4.4 merupakan data hasil pengujian kecepatan dengan perbedaan kategori bayangan. Pada pengujian ini terdapat dua kategori bayangan yaitu tidak ada bayangan dan ada bayangan dengan menggunakan *frame rate* 30 fps. Sesuai dengan data hasil pengujian pada tabel 4.4 perbandingan nilai galat yang dihasilkan dari pengujian tidak ada bayangan dengan ada bayangan menunjukkan bahwa pada pengujian kecepatan tidak ada bayangan, nilai galat yang dihasilkan lebih kecil dari pada pengujian kecepatan dengan ada bayangan. Pada gambar 4.4 merupakan grafik nilai galat

Tabel 4.4: Hasil pengujian perhitungan kecepatan kendaraan dengan perbedaan kategori bayangan

kecepatan	kategori				fps
	tidak ada bayangan		ada bayangan		
	Terdeteksi	galat	Terdeteksi	galat	
20 km/jam	19.5	2.50%	18	10.00%	30
	20.6471	3.24%	17.55	12.25%	
	20.6471	3.24%	14.625	26.88%	
30 km/jam	28.08	6.40%	24.2069	19.31%	
	29.25	2.50%	25.0714	16.43%	
	30.5217	1.74%	26	13.33%	
40 km/jam	36.9474	7.63%	31.9091	20.23%	
	35.1	12.25%	31.9091	20.23%	
	35.1	12.25%	33.4286	16.43%	
50 km/jam	46.8	6.40%	41.2941	17.41%	
	41.2941	17.41%	43.875	12.25%	
	50.1428	0.29%	39	22.00%	

dari pengujian kecepatan dengan perbedaan kategori bayangan. Nilai galat pada grafik tersebut merupakan rata-rata dari nilai galat setiap pengujian kecepatan, dimana berdasarkan data tersebut menunjukkan bahwa pengujian kecepatan dengan ada bayangan menghasilkan nilai galat yang lebih besar dari pada pengujian kecepatan dengan tidak ada bayangan.



Gambar 4.4: Galat hasil pengujian.

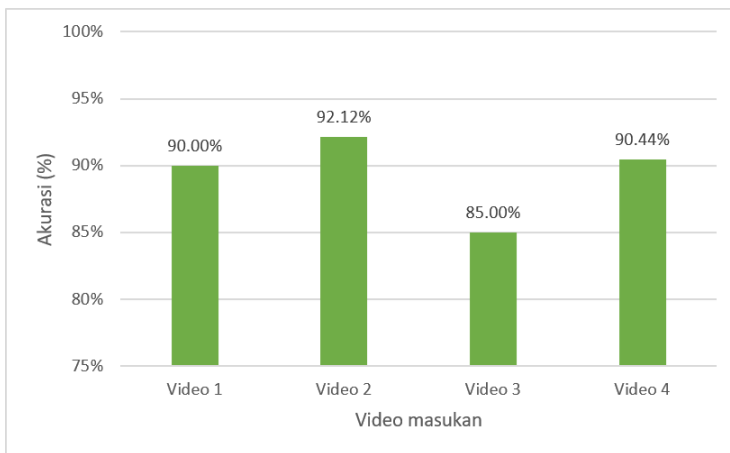
4.3 Pengujian Jenis Kendaraan

Pengujian jenis kendaraan dilakukan untuk mengetahui akurasi sistem dalam membedakan jenis kendaraan yang terdeteksi. Jenis kendaraan yang dideteksi hanya ada dua jenis yaitu mobil dan sepeda motor. Pada tabel 4.5 merupakan hasil pengujian jenis kendaraan terdapat empat video yang diujikan. Keterangan pada tabel 4.5 yaitu MM = Mobil Manual, MT = Mobil Terhitung, SPM = Sepeda Motor Manual, SPT = Sepeda Motor Terhitung. Terhitung merupakan hasil perhitungan jumlah kendaraan oleh sistem dan Manual merupakan hasil perhitungan secara cacah manual.

Tabel 4.5: Hasil pengujian jenis kendaraan

Video	MM	MT	Galat	SPM	SPT	Galat	akurasi
Video 1	22	19	13.64%	88	84	4.55%	90.91%
Video 2	22	24	9.09%	75	70	6.67%	92.12%
Video 3	16	14	12.50%	80	94	17.50%	85.00%
Video 4	13	11	15.38%	107	103	3.74%	90.44%
Rata-rata akurasi							89.62%

Berdasarkan hasil pengujian pada tabel 4.5 menunjukkan nilai galat yang dihasilkam pada pengujian jenis kendaraan mobil dan sepeda motor. Rata-rata galat yang dihasilkan dari deteksi jenis kendaraan mobil yaitu sebesar 12.65% dan untuk sepeda motor sebesar 8.11%. Nilai galat yang didapat dari pengujian ini cukup sedikit jadi sistem ini cukup bagus dalam mendeteksi jenis kendaraan. Terjadi dalam salah deteksi dikarenakan antara objek satu dan objek lainnya jaraknya terlalu berdekatan sehingga dua objek yang terhitung manual tetapi terhitung sebagai satu objek. Berdasarkan tabel 4.5 dan gambar 4.5 tingkat akurasi yang dihasilkan cukup bagus. Pengujian pada video 3 memiliki tingkat akurasi yang paling kecil dari pengujian pada video lainnya, tingkat akurasi yang dihasilkan yaitu sebesar 85%. Rata-rata akurasi hasil penjejakan dalam 4 video ini cukup bagus yaitu sebesar 89.62%.



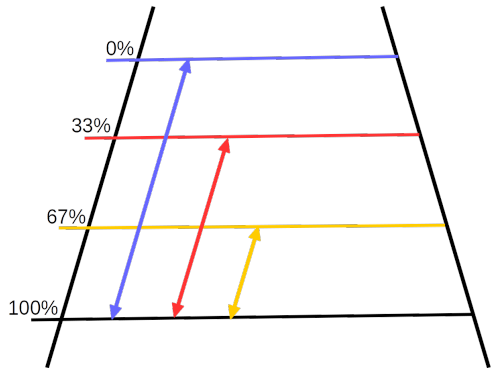
Gambar 4.5: Hasil pengujian jenis kendaraan.

4.4 Pengujian dengan Perbedaan Pemilihan ROI

Pada pengujian ini yaitu pengujian kecepatan dengan perbedaan pemilihan ROI (*Region of Interest*). Pengujian ini dilakukan untuk mengetahui galat yang didapat sistem dalam mendeteksi kecepatan berdasarkan pemilihan *Region of Interest* yang diujikan. *Region of Interest* adalah suatu bagian yang dipilih untuk diproses,

dalam hal ini *Region of Interest* digunakan untuk menentukan batas awal dan batas akhir yang harus ditempuh oleh setiap objek yang terdeteksi. Terdapat tiga *Region of Interest* yang diujikan.

Berdasarkan gambar 4.6, terdapat tiga ukuran *Region of Interest* yang diujikan yaitu ROI 0%-100%, ROI 33%-100%, dan ROI 67%-100%. Pada kondisi nyata, ROI dengan cakupan 0%-100% memiliki ukuran panjang sebesar 650 cm, ROI dengan cakupan 33%-100% memiliki ukuran panjang sebesar 432 cm, dan ROI dengan cakupan 67%-100% memiliki ukuran panjang sebesar 218 cm. Berdasarkan data pengujian pada tabel 4.6 terdapat tiga kecepatan yang diujikan yaitu 20 km/jam, 30 km/jam, dan 40 km/jam.

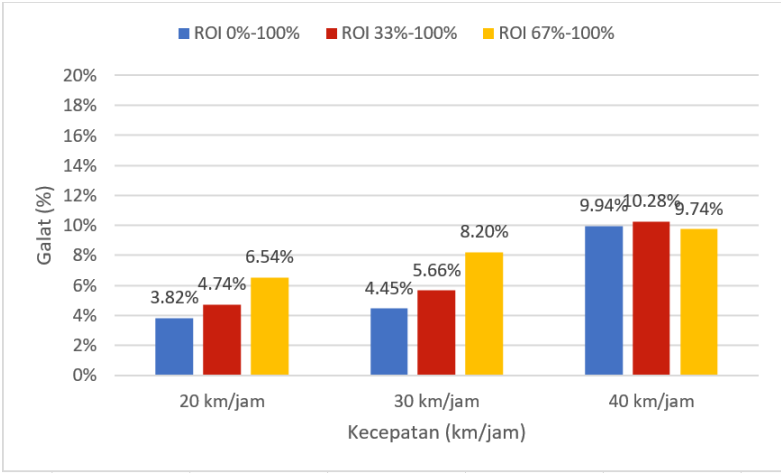


Gambar 4.6: Macam-macam Region of Interest

Tabel 4.6: Hasil pengujian dengan perbedaan pemilihan ROI.

kecepatan km/jam	Pemilihan ROI (Region of Interest)					
	0%-100%		33%-100%		67%-100%	
	terdeteksi	galat	terdeteksi	galat	terdeteksi	galat
20	18.973	5.14%	19.44	2.80%	19.44	2.80%
	19.5	2.50%	18.6624	6.69%	17.9446	10.3%
30	29.25	2.50%	29.16	2.80%	29.16	2.80%
	28.08	6.40%	27.4447	8.52%	25.92	13.6%
40	36.9474	7.63%	35.8892	10.3%	33.3257	16.7%
	36.9474	7.63%	35.8892	10.3%	38.88	2.80%

Data pengujian yang didapat berdasarkan tabel 4.6 menunjukkan bahwa nilai galat terkecil didapat pada pengujian kecepatan 20 km/jam untuk semua jenis ROI. Jika berdasarkan ROI yang diujikan, nilai galat terkecil didapat pada pengujian dengan ukuran ROI 0%-100%. Rata-rata nilai galat yang didapat dari pengujian setiap kecepatan dengan tiga jenis ROI dapat dilihat pada grafik di gambar 4.7. Berdasarkan data pada gambar 4.7 menunjukkan nilai galat terbesar didapat pada pengujian kecepatan 40 km/jam yaitu dengan nilai galat yang didapat dari setiap pengujian ROI sebesar 9.94%, 10.28%, dan 9.74%. Berdasarkan hasil galat yang didapat sesuai data pada gambar 4.7 menunjukkan bahwa sistem memiliki tingkat akurasi terbesar dalam mendeteksi kecepatan jika menggunakan pemilihan ROI 0%-100% dengan ukuran panjang pada kondisi nyata sebesar 650 cm.



Gambar 4.7: Data galat hasil pengujian dengan perbedaan pemilihan ROI.

4.5 Pengujian Pelanggaran Kecepatan

Pengujian kecepatan dilakukan bertujuan untuk mengetahui apakah sistem sudah dapat mengambil citra pengemudi yang melanggar dengan benar atau tidak. Berdasarkan data pada tabel

4.7 yaitu merupakan data pengujian pelanggaran kecepatan dengan data masukan video yang berbeda-beda. Keterangan pada tabel 4.7 yaitu jumlah kendaraan adalah jumlah kendaraan yang melaju dengan kecepatan diatas batas yang terhitung secara manual, tangkap citra 1 dan 2 adalah jumlah citra yang ditangkap oleh setiap kamera jika terdeteksi pelanggaran kecepatan. Sesuai dengan data pada tabel 4.7, data jumlah kendaraan dan jumlah tangkap citra untuk citra 1 dan citra 2 memiliki jumlah yang sama tetapi terdapat kesalahan dalam pengambilan citra pelanggar kecepatan. Kesalahan dalam pengambilan citra pelanggar diakibatkan karna terdapat lebih dari satu objek terdeteksi yang terlalu berdekatan dan juga terdapat objek terdeteksi yang bentuknya tidak pejal sehingga satu objek dapat menjadi dua objek. Berdasarkan hasil dari pengujian ini, tingkat akurasi yang dihasilkan yaitu rata-rata sebesar 97.71%.

Tabel 4.7: Hasil pengujian pelanggaran kecepatan

Video	Batas Kecepatan	Pelanggaran kecepatan			Akurasi
		Jumlah Kendaraan	Tangkap Citra 1 dan 2		
			Benar	Salah	
Video 1	20 km/jam	92	92	0	100.00%
Video 2	20 km/jam	71	69	2	97.18%
Video 3	20 km/jam	82	79	3	96.34%
Video 4	20 km/jam	74	72	2	97.30%
Rata-rata akurasi					97.71%


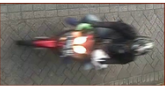




Berdasarkan pada gambar 4.8, pengujian pelanggaran kecepatan dilakukan dengan batas kecepatan 25 km/jam, dimana jika terdeteksi kendaraan yang terhitung kecepatannya melebihi batas tersebut maka sistem akan mengambil citra pelanggar tersebut. Sesuai pada gambar 4.8, telah terdeteksi kendaraan yang melaju melebihi batas kecepatan yaitu dengan kecepatan yang terhitung sebesar 31.9 km/jam dan telah tetangkap citra pelanggar tersebut dari dua posisi kamera.

Data pelanggar yang tertangkap oleh sistem berupa citra pelanggar dari setiap kamera, besar kecepatan, waktu kejadian, dan jenis kendaraan. Setiap data pelanggar yang tertangkap akan disimpan pada sebuah *MySQL Database* di *local server*. Setiap data

yang tersimpan pada database akan ditampilkan pada sebuah web yang dijalankan menggunakan *Apache Web Server*. Pada gambar 4.9 merupakan tampilan web yang telah dibuat, dimana data sesuai pada gambar tersebut menjelaskan bahwa terdapat data pelanggaran yang terdeteksi oleh sistem. Dalam hal ini, dapat dikatakan bahwa aplikasi web yang dibuat sudah dapat menampilkan setiap data pelanggaran kecepatan yang tersimpan pada sebuah database lokal. Berdasarkan data pada gambar 4.8, nilai kecepatan yang terbaca sistem telah dapat terekam pada sebuah *list box*, dimana terdapat dua warna yang menunjukkan bahwa warna merah merupakan daftar kecepatan yang terdeteksi melebihi batas dan warna hijau merupakan kecepatan yang terdeteksi dibawah batas kecepatan. Tepat dibawah *list box* kecepatan terdapat sebuah tombol detail, dimana fungsi dari tombol tersebut adalah untuk melihat detail dari data setiap pelanggaran kecepatan, dimana jika tombol itu ditekan maka akan menampilkan sebuah *dashboard web* seperti pada gambar 4.9.

Pada pengujian pelanggaran kecepatan ini juga terjadi proses pengiriman data pelanggaran ke sebuah *MySQL Database* di *local server*. Pada proses pengujian, setiap data yang terdeteksi melanggar akan disimpan pada sebuah file.txt sebagai penampungan data sementara. Selanjutnya semua data yang tersimpan pada file.txt tersebut dikirim ke database dengan menjalankan program.php yang sudah dikonfigurasi sebelumnya. Terdapat beberapa data yang dikirim ke *Database* yaitu waktu kejadian, Jenis kendaraan, Kecepatan, alamat lokasi citra 1 dan alamat lokasi citra 2. Data yang dikirim ke database hanya data para pelanggaran kecepatan saja.

Pada tabel 4.8 merupakan tabel pada *Database* yang telah dibuat. Berdasarkan data tabel pada gambar tersebut terlihat bahwa setiap data dari pelanggaran kecepatan yang terdeteksi oleh sistem telah dapat masuk ke database. Semua data yang terdapat pada tabel tersebut akan ditampilkan pada sebuah aplikasi web seperti pada gambar 4.9 diatas.

localhost/index/tables.php					
2	2018-05-21 13:22:06	Sepeda Motor	39 km/jam		
3	2018-05-21 13:22:07	Mobil	46.8 km/jam		
4	2018-05-21 13:22:10	Sepeda Motor	41.2941 km/jam		

Gambar 4.9: Percobaan Pelanggaran Kecepatan.

Tabel 4.8: Percobaan Pelanggaran Kecepatan.

waktu	jenis	kecepatan	foto1	foto2
2018-05-30 19:34:21.005641	Sepeda Motor	39 km/jam	images/foto1/1.jpg	images/foto2/1.jpg
2018-05-30 19:34:22.075197	Mobil	46.8 km/jam	images/foto1/2.jpg	images/foto2/2.jpg
2018-05-30 19:34:24.256760	Sepeda Motor	41.2941 km/jam	images/foto1/3.jpg	images/foto2/3.jpg
2018-05-30 19:34:30.165752	Sepeda Motor	29.25 km/jam	images/foto1/4.jpg	images/foto2/4.jpg
2018-05-30 19:34:32.841184	Sepeda Motor	29.25 km/jam	images/foto1/5.jpg	images/foto2/5.jpg
2018-05-30 19:34:52.968803	Sepeda Motor	28.08 km/jam	images/foto1/6.jpg	images/foto2/6.jpg
2018-05-30 19:34:58.223543	Sepeda Motor	39 km/jam	images/foto1/7.jpg	images/foto2/7.jpg
2018-05-30 19:34:59.508533	Mobil	25.0714 km/jam	images/foto1/8.jpg	images/foto2/8.jpg
2018-05-30 19:35:02.023200	Sepeda Motor	33.4286 km/jam	images/foto1/9.jpg	images/foto2/9.jpg
2018-05-30 19:35:04.768832	Sepeda Motor	33.4286 km/jam	images/foto1/10.jpg	images/foto2/10.jpg
2018-05-30 19:35:12.980277	Sepeda Motor	29.25 km/jam	images/foto1/11.jpg	images/foto2/11.jpg
2018-05-30 19:35:26.576137	Sepeda Motor	30.5217 km/jam	images/foto1/12.jpg	images/foto2/12.jpg
2018-05-30 19:35:30.022941	Sepeda Motor	31.9091 km/jam	images/foto1/13.jpg	images/foto2/13.jpg
2018-05-30 19:35:31.916279	Sepeda Motor	25.0714 km/jam	images/foto1/14.jpg	images/foto2/14.jpg
2018-05-30 19:35:33.883156	Sepeda Motor	36.9474 km/jam	images/foto1/15.jpg	images/foto2/15.jpg
2018-05-30 19:35:36.111295	Sepeda Motor	28.08 km/jam	images/foto1/16.jpg	images/foto2/16.jpg
2018-05-30 19:35:39.015325	Sepeda Motor	28.08 km/jam	images/foto1/17.jpg	images/foto2/17.jpg
2018-05-30 19:35:40.687829	Sepeda Motor	33.4286 km/jam	images/foto1/18.jpg	images/foto2/18.jpg
2018-05-30 19:35:46.399143	Sepeda Motor	29.25 km/jam	images/foto1/19.jpg	images/foto2/19.jpg
2018-05-30 19:35:47.319030	Sepeda Motor	28.08 km/jam	images/foto1/20.jpg	images/foto2/20.jpg

4.6 Durasi Pengiriman Data Pelanggaran Kecepatan

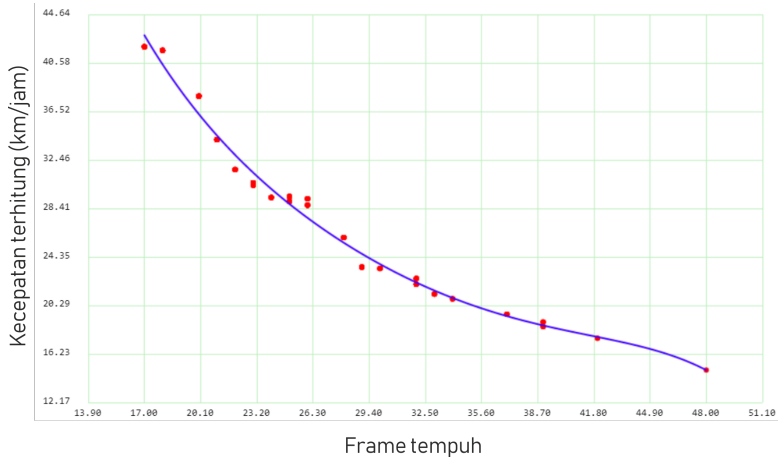
Pada pengujian ini dilakukan agar mengetahui durasi waktu pengiriman data pelanggaran kecepatan dari sistem ke database. Untuk mengetahui durasi pengiriman yaitu dengan mengurangi waktu terima dan waktu kirim. Waktu kirim yaitu waktu sebelum menjalankan proses pengiriman ke database. Waktu terima adalah waktu saat data pelanggaran kecepatan sudah masuk ke database.

Berdasarkan data pada tabel 4.9, merupakan data hasil pengujian durasi pengiriman data pelanggaran kecepatan. Pada percobaan ini dilakukan 15 kali pengujian pengiriman data ke *database* berdasarkan data pelanggaran kecepatan yang terdeteksi oleh sistem. Tingkat ketelitian waktu yang digunakan pada pengujian ini yaitu hingga *milisecond*. Berdasarkan data waktu yang didapat dari waktu kirim hingga waktu terima, durasi pengiriman yang tercepat yaitu sebesar 0.014 detik. Rata-rata durasi pengiriman data saat terjadi pelanggaran kecepatan ke *database* sebesar 0.0147 detik. Dalam hal ini dapat disimpulkan bahwa waktu yang dibutuhkan untuk pengiriman data ke *database* yang terdapat pada *local server* sangatlah cepat yaitu kurang dari satu detik.

Tabel 4.9: Pengujian durasi pengiriman data pelanggaran kecepatan ke database

No	Tanggal	Waktu kirim	Waktu terima	Durasi pengiriman (detik)
1	2018-05-30	19:34:22.061	19:34:22.075	0.014
2	2018-05-30	19:34:24.241	19:34:24.256	0.015
3	2018-05-30	19:34:30.150	19:34:30.165	0.015
4	2018-05-30	19:34:32.827	19:34:32.841	0.014
5	2018-05-30	19:34:52.953	19:34:52.968	0.015
6	2018-05-30	19:34:58.208	19:34:58.223	0.015
7	2018-05-30	19:34:59.494	19:34:59.508	0.014
8	2018-05-30	19:35:02.008	19:35:02.023	0.015
9	2018-05-30	19:35:04.754	19:35:04.768	0.014
10	2018-05-30	19:35:12.965	19:35:12.980	0.015
11	2018-05-30	19:35:26.561	19:35:26.576	0.015
12	2018-05-30	19:35:30.006	19:35:30.022	0.016
13	2018-05-30	19:35:31.902	19:35:31.916	0.014
14	2018-05-30	19:35:33.868	19:35:33.883	0.015
15	2018-05-30	19:35:36.096	19:35:36.111	0.015
Rata-rata durasi pengiriman				0.0147

4.7 Pengujian Kecepatan dengan Menggunakan Persamaan Regresi



Gambar 4.10: Grafik regresi polinomial orde 5.

Pada gambar 4.10 merupakan grafik dari regresi polinomial orde 5 dengan data masukan jumlah *frame* dan kecepatan. Berdasarkan gambar 4.10, kecepatan terhitung merupakan nilai kecepatan yang didapat dari perhitungan manual dengan menggunakan media player sebagai *tool* untuk mengetahui waktu tempuh suatu kendaraan dalam satuan *millisecond* dan *frame* tempuh merupakan jumlah frame yang ditempuh suatu kendaraan saat terdeteksi oleh sistem. Persamaan 4.2 merupakan persamaan regresi polinomial orde 5 yang didapat sesuai data masukan yang digunakan.

$$\begin{aligned}
 k = & 1.8325028463625927 \times 10^2 \times f^0 - 1.8688827198986765 \times 10^1 \times f^1 \\
 & + 9.9519226333690303 \times 10^{-1} \times f^2 - 2.9245826901324750 \times 10^{-2} \times f^3 \\
 & + 4.4723609676484185 \times 10^{-4} \times f^4 - 2.7628994756743460 \times 10^{-6} \times f^5
 \end{aligned}
 \tag{4.2}$$

Dengan,

$$\begin{aligned}
 k &= \text{Kecepatan} \\
 f &= \text{Frame tempuh}
 \end{aligned}$$

Dari persamaan 4.2 yang didapat, maka dilakukan pengujian dengan mengubah rumus perhitungan kecepatan pada sistem dengan persamaan 4.2. Pada tabel 4.10 merupakan hasil dari proses perhitungan kecepatan menggunakan persamaan regresi yang sudah didapat. Berdasarkan data yang didapat pada tabel 4.10 menunjukkan bahwa nilai rata-rata galat yang dihasilkan yaitu sebesar 2.11%.

Tabel 4.10: Pengujian kecepatan dengan pendekatan regresi polinomial orde 5

Kecepatan terhitung (km/jam)	Frame tempuh	Kecepatan terdeteksi (km/jam)	Galat
40.20618557	18	40.46	0.63%
48.44720497	15	48.67	0.46%
42.31464738	17	42.9	1.38%
30.99337748	24	30.04	3.08%
25.46245919	29	24.61	3.35%
28.85326757	25	28.78	0.25%
24.375	30	23.74	2.61%
20.93023256	35	20.36	2.72%
41.4159292	18	40.46	2.31%
26.20380739	28	25.54	2.53%
34.51327434	21	34.51	0.01%
23.70820669	30	23.74	0.13%
35.02994012	20	36.3	3.63%
22.11720227	33	21.54	2.61%
33.71757925	22	32.89	2.45%
26.35135135	28	25.54	3.08%
39.13043478	19	38.27	2.20%
25.49019608	29	24.61	3.45%
27.85714286	26	27.62	0.85%
20.3125	37	19.4	4.49%
Rata-rata galat			2.11%

BAB 5

PENUTUP

5.1 Kesimpulan

Dalam penelitian ini, telah diimplementasikan serangkaian prosedur untuk menentukan pelanggaran kecepatan kendaraan dengan proses penjejakan menggunakan algoritma *matching based*. Berdasarkan hasil pengujian, tingkat akurasi sistem dalam menghitung kecepatan yaitu mencapai 97.01 % pada kondisi tidak ada bayangan dan pada kondisi ada bayangan, tingkat akurasi dalam menghitung kecepatan yaitu mencapai 83.86 %. Hal tersebut terjadi karena pencahayaan yang dihasilkan oleh cahaya matahari mengakibatkan adanya bayangan tambahan pada objek, sehingga proses perhitungan kecepatan menjadi kurang maksimal. Dalam menangkap citra pengendara yang terdeteksi melaju dengan kecepatan diatas nilai batas, sistem ini sudah cukup baik dalam mengambil citra pengendara yang melanggar kecepatan dari dua posisi kamera. Sehingga tingkat akurasi dalam memacu tiap kamera untuk mengambil citra pelanggar kecepatan sebesar 97.71%, dimana pengujiannya dilakukan secara berkali kali dengan video masukan yang berbeda. Data dari setiap pelanggar kecepatan disimpan di sebuah database pada *local server* dan berdasarkan hasil pengujian, durasi rata-rata yang dibutuhkan untuk pengiriman ke database yaitu sebesar 0.0147 detik.

5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, terdapat beberapa saran sebagai langkah lanjutan yaitu sebagai berikut :

1. Perbaikan pada algoritma penjejakan kendaraan. Algoritma penjejakan yang digunakan pada sistem ini yaitu *matching-based* dengan data masukan nilai *euclidean distance* setiap kendaraan antar *frame* sehingga jika banyak objek yang terdeteksi kerap terjadi kesalahan karna banyak data yang dibandingkan untuk setiap objek.
2. Pengambilan data dengan posisi kamera yang berbeda un-

tuk mendapatkan hasil perhitungan kecepatan yang maksimal. Dalam penelitian ini karna faktor infrastuktur yang menyebabkan posisi kamera yang digunakan sebagai deteksi kecepatan tidak dapat diubah, dimana cakupan luas citra yang didapat di kondisi nyata sebesar 7.28 x 5.83 meter, sehingga proses perhitungan kecepatan kurang bisa maksimal.

3. Sistem yang diajukan masih membutuhkan pengujian pada kondisi lingkungan yang berbeda-beda, seperti hujan, berawan, atau berkabut.
4. Menambahkan sistem deteksi plat nomor kendaraan. Sehingga jika terdeteksi pelanggar kecepatan, data dari citra setiap pelanggar kecepatan dapat di proses secara otomatis agar bisa mendapatkan data plat nomor dari kendaraan yang melanggar.

DAFTAR PUSTAKA

- [1] “Menhub: Cegah kecelakaan, kurangi kecepatan kendaraan!” Terakhir diakses pada tanggal 21 Maret 2018. (Dikutip pada halaman i, 1).
- [2] P. Hartoto, “Sistem deteksi kecepatan kendaraan bermotor pada real time traffic information system,” bachelor thesis, Institut Teknologi Sepuluh Nopember, January 2012. (Dikutip pada halaman 1, 15).
- [3] N. H. Tsani, “The implementattion of vehicle speed detection using webcam with frame difference method,” bachelor thesis, Telkom University, August 2017. (Dikutip pada halaman 6).
- [4] R. M. Tani, “Perancangan antarmuka ip-cam wifi robot,” bachelor thesis, UNSRAT. (Dikutip pada halaman 6).
- [5] D. W. Wicaksono and B. Setiyono, “Speed estimation on moving vehicle based on digital image processing,” February 2017. (Dikutip pada halaman 6, 13).
- [6] E. R. F. A. Putra, Lilina, and K. Gunadi, “Aplikasi automatic image stiching pada kumpulan gambar dalam satu scene.” Terakhir diakses pada tanggal 12 April 2018. (Dikutip pada halaman 7).
- [7] W. A. Hoff, Computer Vision. 1610 Illinois St. Golden, Colorado 80401-1887: DAQRI, 2016. (Dikutip pada halaman 9).
- [8] D. Kriegman, Homography Estimation. CSE 252A, 2007. (Dikutip pada halaman 9).
- [9] A. M. A. M. AKBAR, “Traffic ip camera for counting four-wheeled vehicle using area of pixels method,” bachelor thesis, Institut Teknologi Sepuluh Nopember, April 2016. (Dikutip pada halaman 9, 11, 15, 44).
- [10] Z. Yi and F. Liangzhong, “Moving object detection based on running average background and temporal difference,” in 2010 IEEE International Conference on Intelligent Systems and

- Knowledge Engineering, pp. 270–272, Nov 2010. (Dikutip pada halaman 10).
- [11] W. Huang, Y. Kang, and S. Zheng, “An improved frame difference method for moving target detection,” in 2017 Chinese Automation Congress (CAC), pp. 1537–1541, Oct 2017. (Dikutip pada halaman 11, 12).
 - [12] S. B. PRAKOSO, “Pengembangan sistem perhitungan jumlah kendaraan memanfaatkan citra aerial yang diambil dengan pesawat tanpa awak,” bachelor thesis, Institut Teknologi Sepuluh Nopember, March 2018. (Dikutip pada halaman 13).
 - [13] M. Mohanty, M. R. Asghar, and G. Russello, “*2dcrypt* : Image scaling and cropping in encrypted domains,” IEEE Transactions on Information Forensics and Security, vol. 11, pp. 2542–2555, Nov 2016. (Dikutip pada halaman 13).
 - [14] Y. Cao, X. Yun, T. Zhong, and X. Huang, “Research on vehicle detection and tracking algorithm based on the methods of frame difference and adaptive background subtraction difference,” vol. 133, December 2016. (Dikutip pada halaman 14, 51).
 - [15] I. K. E. Purnama, A. Zaini, B. N. Putra, and M. Hariadi, “real time vehicle counter system for intelligent transportation system,” November 2009. (Dikutip pada halaman 14, 50).
 - [16] “Technical specification of bullet network camera ib8354-c.” Terakhir diakses pada tanggal 28 Maret 2018. (Dikutip pada halaman 28).
 - [17] “Technical specification of bullet network camera sno-5084r.” Terakhir diakses pada tanggal 28 Maret 2018. (Dikutip pada halaman 28).
 - [18] N. B. A. Mustafa, F. Bakri, and S. K. Ahmed, “Identification of image angle using projective transformation: Application to banana images,” in 2014 IEEE REGION 10 SYMPOSIUM, pp. 408–413, April 2014. (Dikutip pada halaman 32).

- [19] D. C. Luvizon, B. T. Nassu, and R. Minetto, “A video-based system for vehicle speed measurement in urban roadways,” vol. 18, pp. 1393–1404, June 2017. (Dikutip pada halaman 33).
- [20] “Operations on arrays.” Terakhir diakses pada tanggal 17 April 2018. (Dikutip pada halaman 36).
- [21] N. Wakhidah, “Ekstraksi ciri plat nomor mobil menggunakan moment pusat,” vol. 8, pp. 67–75, January 2013. (Dikutip pada halaman 48).

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Aldiansyah Ramadlan, lahir pada 27 Januari 1996 di Surabaya, Jawa timur. Penulis lulusan dari SMP Wachid hasyim 4 Surabaya pada tahun 2011 kemudian melanjutkan pendidikan ke SMA Negeri 21 Surabaya hingga akhirnya lulus pada tahun 2014. Penulis kemudian melanjutkan pendidikan Strata satu ke Departemen Teknik Komputer ITS Surabaya dibidang studi Telematika. Selama masa kuliah penulis aktif menjadi Asisten laboratorium B401 Komputasi Multimedia dan pernah menjabat sebagai koordinator dalam bidang *Internet of Things Development*. Selama masa kuliah, penulis aktif di Microsoft sebagai *Microsoft Student Partner* Indonesia dalam membagi keilmuan seputar teknologi kepada mahasiswa kampus, selain itu penulis aktif mengikuti ajang serta kompetisi teknologi tingkat mahasiswa ataupun umum. Penulis banyak menghabiskan waktu untuk mengeksplor isu-isu terkait teknologi dunia. Untuk kepentingan kepada penulis, dapat menghubungi aldiansyahxramadlan@gmail.com

Halaman ini sengaja dikosongkan